NOV/DEC 86
VOL. 3 NO. 1
$3.00

BULK RATE
U.S. Postage
PAID
Colton, OR 97017
Permit No. 51

POSTMASTER:
PLEASE EXPEDITE DELIVERY
DATED MATERIAL

They Are Still Around, Write them For Prices

AMERICA'S LARGEST TIMEX SINCLAIR MAGAZINE

TIME DESIGN

MAGAZINE

"What's All The Commotion About?"

"IT'S TDM's 2nd ANNIVERSARY ISSUE!"

IT'S HERE!!

QL TRIVIA-QLUE

HERE IS GREAT NEW COMBINATION GAME!
IF YOU LIKED QLUE AND TRIVIA, THEN THIS IS FOR YOU!
WE HAVE COMBINED THESE TWO GREAT THINKING MANS' GAMES
IN ONE GREAT PACKAGE!!

13 ROOMS - 13 WEAPONS - 13 SUSPECTS
THAT'S OVER 2100 POSSIBLE OUTCOMES!
AND NOW - OVER 750 TRIVIA QUESTIONS!
QUESTIONS COVER EVERY CONCEIVABLE SUBJECT!
RELIGION - HISTORY - SCIENCES - GEOGRAPHY - ASTRONONMY
YOU'VE NEVER SEEN QUESTIONS LIKE THESE!

Also included, at no extra cost, your choice of
FOOTBALL FORECASTER or LIFE EXPECTANCY FORECASTER
(Specify which you would like when you order.)

WE THINK YOU'LL LIKE THIS ONE!
OUR PRICE? ONLY **$19.95** +$2ph

AVAILABLE EXCLUSIVELY FROM:
RMG ENTERPRISES

ANNOUNCING

SOUNDESIGN 2068

A NEW PROGRAMMING UTILITY FOR YOUR 2068!

This little package allows you to design your own sound effects or musical effects for your programs simply and easily. Completely menu driven, using only 3 keys, you can design a sound, print out the necessary program line (generated by our program) and insert it into your program.

The program will let you test out sounds, change volume & pitch, open or close the sound channels - all with only 3 keys!

Sound simple? IT IS!
We want you to be able to use our utilities, and the simpler we make them, the easier they are to use and the more likely you will use and recommend them!

This one it NEW from ARROW SOFTWARE, a company we hope to hear a lot from in the months ahead!

AT ONLY **$12.95**+$1.50ph
THIS ONE IS A MUST FOR ALL PROGRAMMERS!
IT WILL SAVE YOU MORE THAN THAT IN TIME ALONE!

SOUNDESIGN
IS AVAILABLE EXCLUSIVELY FROM

RMG ENTERPRISES

MAKE CHECKS OR MONEY ORDERS OUT TO:
RMG ENTERPRISES
1419 1/2 7TH STREET
OREGON CITY, OR 97045
503/655-7484
10 AM TILL 10 PM TUE-SAT

ANNOUNCING!

SEKTOR 2068

FOR THE FD68 USER

Finally! A full-featured sector editor for use with your 2068 and AERCO FD68 disk system. Very user friendly! Many features that you would not expect at this price!

Some of the features included are:

READ DISK SECTORS - WRITE SECTORS - CORRECT SECTORS
DELETE/ERASE FILES - COMPARE BLOCKS - COPY SCREENS
TRANSLATE IBM CLUSTERS - MSDOS (2 VERSIONS)
- WORKS WITH RP/M AND MORE!

All of this at a price that all FD68 users can afford:
ONLY **$15.00**+2ph

Available on 5 1/4" disk, 3" disk or cassette.
Please add $5 for 3" disk version.
Complete with very thorough documentation.

AVAILABLE EXCLUSIVELY FROM:
RMG ENTERPRISES

OUR NEW CATALOG IS COMING!

WE WANT YOU TO GET YOUR COPY!

OUR NEW CATALOG IS BIGGER AND BETTER THAN EVER!

WE'VE ADDED NEW PRODUCTS IN ALL OF OUR OLD LINES
AND ARE ADDING NEW LINES OF MERCHANDISE.

OUR OLD LINES INCLUDE:
TS 2068 * TS 1500 * TS 1000/ZX81 * QL

OUR NEW LINES INCLUDE:
COMPUTER FORMS & PAPER PRODUCTS * DISK DRIVES
DISK DRIVE CASES & POWER SUPPLIES * DISK DRIVE CABLES
BLANK DISKS * BLANK VIDEO TAPES * CB RADIOS
MUCH MORE TO COME!!

BECAUSE OF THE HIGHER COSTS OF PRODUCING AND MAILING A NEW CATALOG, WE ARE FORCED TO ASK, THAT, IF YOU ARE A SERIOUS CUSTOMER, AND WOULD LIKE TO RECEIVE ONE OF OUR NEW CATALOGS, YOU SEND US $2.00 TO COVER THESE COSTS. IN RETURN, WE WILL DEDUCT THE $2.00 OFF YOUR FIRST REGULAR ORDER! THAT WAY YOU DON'T ACTUALLY LOSE ANYTHING AND HAVE A LOT OF SAVINGS TO GAIN!

JUST FILL IN THE INFORMATION BELOW AND SEND THIS FORM ALONG WITH YOUR CHECK OR MONEY ORDER FOR $2.00 TO THE ADDRESS BELOW!

AND THAT'S NOT ALL!
IF YOU ENCLOSE 12 LEGAL SIZED S.A.S.E.s WITH THIS FORM YOU WILL BE SURE TO RECEIVE OUR MONTHLY SPECIALS, CLOSE-OUTS, NEW PRODUCT NOTICES AND CATALOG UPDATES. YOU WILL ALSO RECEIVE A COUPON GOOD FOR A $5.00 DISCOUNT ON YOUR NEXT CATALOG ORDER!

THAT WAY IT ACTUALLY COSTS YOU NOTHING!

DON'T MISS OUT! ORDER TODAY!

NOV/DEC 86 VOL. 3 NO. 1

TIME DESIGNS MAGAZINE

FOR ALL TIMEX AND SINCLAIR COMPUTERS

TIME DESIGNS MAGAZINE CO.
29722 Hult Rd. • Colton, Oregon 97017
(503) 824-2658

TIME DESIGNS MAGAZINE is published bi-monthly and is Copyright © 1986 by the **Time Designs Magazine Company,** Colton, Oregon 97017. All rights reserved.

Editor: Tim Woods

Assistant Editor: Stephanie Woods

Editorial Assistant/Production: D.L. Woods

Photography:
(unless otherwise noted): Thomas Judd
Printing by; Toad'l Litho Printing and Comp., Oregon City, Oregon 97045

SUBSCRIPTIONS: $15 a year for six issues (US funds only). No extra charge to Canadian subscribers. All other countries please write for information on air mail rates.

CUSTOMER SERVICE: Customer satisfaction is our goal. For subscription service problems please write or call TIME DESIGNS.

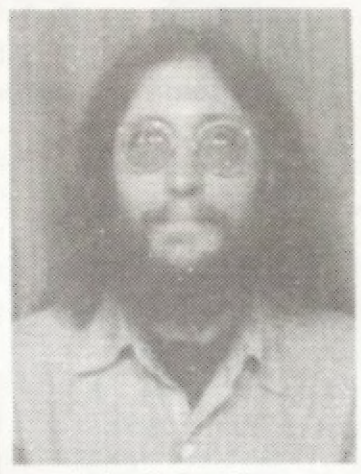
CHANGE OF ADDRESS: Write or call to prevent delay of sevice,

The Folks Who Bring You TIME DESIGNS

Tim Woods
Editor

Stephanie Woods
Assistant Editor

Tim Woods and Tom Judd,
Staff Photographer

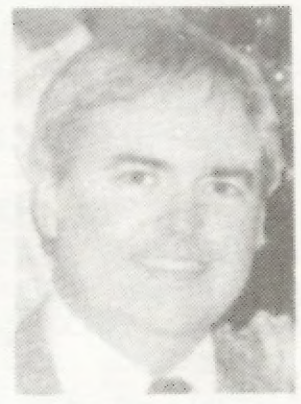


Michael E. Carver

Paul Bingham

Mike de Sosa

Joe Williamson



Duncan Teague

Wes Brzozowski

D. Hutchinson

Syd Wyncoop

Stan Lemke

Earl V. Dunnington

This being our 2nd Anniversary Issue and the start of our 3rd year of publication, I'm going to pass up the usual column featured here, and introduce you to some folks responsible for putting out TIME DESIGNS six times a year. Most of you know this is a "family" business, which not only employs both my wife and I, but also other family members and friends on a part time basis. TDM really wouldn't be possible without our great contributors...I feel the very best around. Some of them have been with us since Volume One.

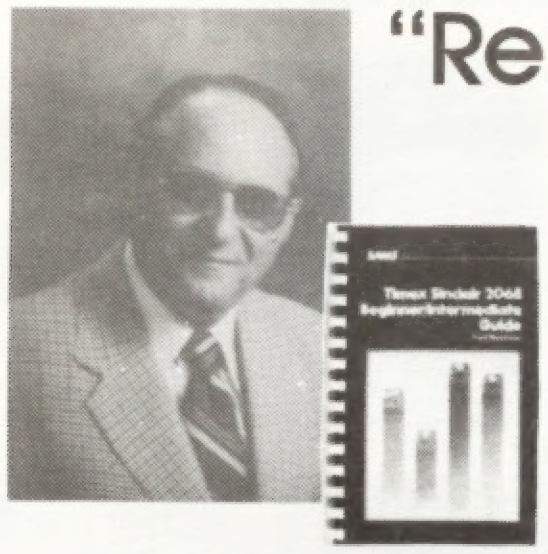
Above you will find photo's of some of these people who you have read about, but this time you can tie a picture to a name. It's all in fun, and at the same time gives them some deserving recognition. (I've even included my own "mug shot" for what it's worth.) There are many others who aren't pictured above, who are also regulars to our pages, such as: Tim Stoddard, Warren Fricke, Bill Ferrebee, Charles E. Goyette, Dick Wagner, Dennis Jurries, Dennis Silvestri, R. Lussier (as well as several others). We'll have to get them next time.

I look forward to working with everyone for the next six issues of TDM, and serving you our readers with the magazine "written by Sinclair enthusiasts---for Sinclair enthusiasts". I also want to wish our writers, their families, as well as our entire readership... Happy Holidays!

-Tim Woods

"Remember Back When..."

by
Fred Blechman



Tim Woods has asked me to reminisce about the early days of the Timex Sinclair microcomputers. Ah, nostalgia time! The good ol' days...or were they?

When you think about it, the "good ol' days" of the Timex computers only go back to April 1982. That was when Timex Computer Corporation, a wholly-owned subsidiary of the Timex Watch Company, announced to a stunned press that it had made an agreement with Sinclair Research Ltd. (England) to produce and market the Timex Sinclair 1000, Timex's version of Sinclair's ZX81...and it was going to sell at 150,000 Timex North American retail outlets for only $99! That didn't happen, since most stores that sold Timex watches decided not to try to sell computers...but it was sure exciting to think about!

But the genesis of the TS 1000 goes somewhat further back to when "Uncle Clive" Sinclair shocked the computer world in early 1980 by announcing the first under-$200 computer, the ZX80. This was an immediate hit in England and came to the U.S., mail-order only, in late 1980. This was followed by the ZX81, which was actually manufactured in Scotland by Timex. It sold for $150 assembled, or $100 in kit form. The ZX81 quickly became the largest-selling computer in the world.

I got my first ZX81 in early 1982. I ordered a kit for $100, but they had more assembled units than kits, so they apologized for sending me an assembled unit!

I had already cut my computing teeth on a Radio Shack TRS-80 Model I 4k with Level I BASIC, which I had upgraded to 16k with Level II BASIC language. I had written one book for Hayden Publishing ("Programs For Beginners On The TRS-80") and many magazine articles, so the ZX81 was not my first micro...but it quickly got my attention.

It was FUN to program the ZX81 in Sinclair BASIC, which was much more powerful than the TRS-80 Level I BASIC. Because there was only 1k of RAM, and much of that was devoted to the screen, there was not much memory left for a program...making the challenge much greater. The graphics were limited, but easy to use.

I started writing articles about the ZX81 and the Timex Sinclair 1000. Since they were identical, except that the TS 1000 had a 2k RAM instead of 1k, everything I did with the ZX81 worked on the TS1000. Furthermore, by adding the 16k RAMpack to a ZX81, it was the same as a TS 1000 with a RAMpack. In fact, I never did get a TS 1000. By the time they were available, I had two ZX81s with RAMpacks!

I recall the difficulty in getting a printer in the early days, before the Timex Sinclair 2040. Sinclair put one out in England for about $100. I don't even remember what they called it, but it put out so much radio-frequency interference that the FCC banned it in this country. It used an electrostatic process that vaporized a thin aluminum coating to expose the black surface on a carbon-coated paper roll. I ordered one of these little printers from Gladstone Electronics, via Canada. The import paperwork, shipping and tariff cost about $35! It was strange, but gave an acceptable 32-column printout that duplicated every dot on the screen. To do that on many of today's micros takes special graphic screen dump programs!

My first ZX/TS-oriented article was in the Sept/Oct 1982 issue of SYNC Magazine. I subsequently wrote 20 other articles covering the ZX81, TS 1000, TS 1500, TS 2068, Spectrum, and QL for other magazines...Electronic Fun, CES Daily, Microcomputing, TODAY (CompuServe), Timex Sinclair User, Computer Shopper, Computers & Electronics, Computer Trader and Modern Electronics. My last "Timex" article was a 7-page hands-on review of the Sinclair QL in the June 1985 issue of Modern Electronics.

Along the way, I wrote the book, "Timex Sinclair 2068 Beginner/Intermediate Guide" for Howard W. Sams. It is now out of print, but available from the E. Arthur Brown Company. They also sell my friend Jeff Mazur's book, "Timex Sinclair 2068 Intermediate/Advanced Guide", also published by Sams, which picks up where mine leaves off. Writing that book was a real challenge, since I didn't have a TS 2068 Personal Color Computer! Dan Ross, the man running Timex Computer at the time, made an arrangement for Jeff and me to each have a Sinclair Spectrum, on which the TS 2068 was based. We also got some advance information, but had to make some educated guesses. Sue Mahoney and George Grimm at Timex were very helpful...thanks, wherever you are today...

The real irony was that my completed book manuscript went to the publisher in early August of 1983... and later that same day Federal Express delivered the first TS 2068 I had ever seen! Luckily, after checking out the actual 2068, I only had to change one paragraph in my manuscript.

As it turned out, the computer had been delayed so long my book hit the streets before the computer...and Timex closed down the computer division just a few months later. What a shame! A great little computer caught in a web of bad engineering and marketing decisions.

In their defense, Timex management had a host of problems with the real value of a "home computer" being challenged, and price wars created by the competition forcing profits too low. Add the unreality of trying to effectively sell a device as complex and unfriendly as a computer in drug stores, and the stage was set for repercussion. Timex was not the only micro manufacturer to fall on bad times. It just seems, however, that if they had "hung in there" about another six months, the superior features of the TS 2068 would have become known.

What have I done since? Well, I've had other computers in my collection...TRS-80 Model III (two of those), TRS-80 Model 4P (two of those), Coleco ADAM, Radio Shack MC-10 Microcolor Computer, Sanyo MBC 555-2, Apple IIc, and just recently got an IBM PC/XT clone.

I've written three more books since my Timex 2068 book, and over 200 computer-related magazine articles. "The ADAM Beginner & Intermediate Guide", a book written for Sams, was cancelled after acceptance and editing, due to the fall of the ADAM. My "Sanyo Beginner & Intermediate Guide" and "Apple IIc - An Intelligent Guide" were published by CBS Computer Books, just before they

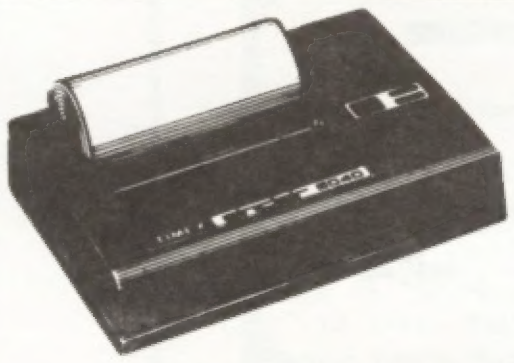
abandoned the computer book market. I've personally sold over 1000 copies of the Sanyo book, since like the Timex machines...it has many devoted users.

I hated the Apple IIc, the ADAM was "unspeakable", but I love the Sanyo! It offers the ease and power of programming remindful of the TS 2068, but with two built-in disk drives and 48,000 pixels on the screen (640 x 200), each in any of eight colors! Wow!

Thank goodness I've got my Amway Emerald Direct Distributorship to support my computerholic tendency! Also, well over 1000 Amway Product Distributors have purchased my $100 "AMBIZ-PAK" of 10 programs for the IBM PC/Clones, TRS-80 Models III/4/4P/4D, and the Sanyo MBC 550 series.

I'm pretty much out of the Timex environment now, and will not be writing about the QL. However, I can well understand the fascination and dedication many of you hold for the Sinclair and Timex machines. As I recall, at one time Timex used ads with the slogan "The POWER is in your hands!" May the POWER be with you...

L E T T E R S



Nowak's Letter Gets Response

Editor's note: Our mail box has been quite full the past two months due to a letter and request we published in the Sept/Oct '86 issue of TDM, on page 3. To paraphrase Mr. Nowak's letter, he requested a short program or routine that would by-pass the monitor and dump directly to the printer for doing simple computations. While the answer appears to be simple, such as the use of the LPRINT command, here is what some of our readers came up with. And thanks to everyone who took the time to write.

Dear Tim,

In the Sept/Oct 1986 issue Michael J. Nowak asked for a way to have the 2068 print to the printer instead of the screen. One simple method, which will work in the immediate mode or as a program line, is: OPEN #2, "P" The "#2" part refers to PRINT and LIST commands. The "P" refers to the 2040 printer ("S" would mean screen in this syntax). Hence, PRINT or LIST will subsequently go to the printer instead of the screen. LLIST, LPRINT, INPUT, and lower screen messages will still appear on the screen. CLOSE #2 gets things back to normal.

Opening and closing files in this way was not mentioned in the 2068 User Manual (more Timex unfinished business), but it can be a useful feature. Listing #1 is an example in which channel #4 is used to give a screen or printer option for the output. I chose #4 because #1, #2, and #3 are reserved for INPUT, PRINT/LIST, and LPRINT/LLIST commands, respectively. It's worth experimenting with!

Sincerely,

Larry Dietrich
Blanca, CO

```
 100 REM EXAMPLE OF DEVICE INDEPENDENT OUTPUT
 110 LET GETKEY=1000
 120 PRINT "Output to Screen or Printer?       (Press S
or P)"''
 130 GO SUB GETKEY
 140 IF I$<>"S" AND I$<>"s" AND I$<>"P" AND I$<>"p" THEN
GO TO 130
 150 OPEN #4,I$
 200 REM BODY OF PROGRAM
 210 FOR L=1 TO 10
 220 PRINT #4;TAB (L<10);L;" squared = ";L*L
 230 NEXT L
 240 PRINT '"DONE": REM THIS PRINTS TO SCREEN
 250 STOP
1000 REM GETKEY SUB
1010 LET I$=INKEY$
1020 IF I$="" THEN GO TO 1010
1030 RETURN
```

```
 1 squared = 1
 2 squared = 4
 3 squared = 9
 4 squared = 16
 5 squared = 25
 6 squared = 36
 7 squared = 49
 8 squared = 64
 9 squared = 81
10 squared = 100
```

LISTING 1

To the Editor,

In response to Mr. Nowak's letter in TDM Sept/Oct '86 issue--the enclosed program works well on the 2068 and TS 1000 (using the proper "to the power" symbol). It can be enhanced by putting in an entry counter with C$ and B tabbed to other locations and/or reversing the position of C$ and B. The base program is:

```
10 INPUT C
20 PRINT C or LPRINT C
30 GOTO 10
```

The value of C is calculated and printed (Lprinted) as a single value. The entry prompted by an L cursor is not printed (Lprinted) and is "lost".

W.B. Gray Jr.
West Caldwell, NJ

```
  1 REM ......Evaluators.......
  2 REM ......LINES 1,2,3,4,51,
100,101 NOT NECESSARY..........
  3 LPRINT "INPUT","ANSWER"
  4 LPRINT "......","......"
 20 INPUT C$
 30 IF C$="0" THEN GO TO 100
 40 LET B=VAL C$
 50 LPRINT C$;TAB 15;"=";B
 51 LPRINT "-------------------
------------"
 60 GO TO 20
100 LPRINT
101 LPRINT "********************
************"
102 STOP
```

```
3+8= 11
------------------------------
5*7= 35
------------------------------
8+10= 45
------------------------------
8/9= 5
------------------------------
(8/4)+11= 12.25
------------------------------
```

To the Editor,

Regarding Mr. Nowak's letter: There is a very simple solution to this, but it has two small drawbacks. After turning on the 2068, type in:
POKE 26692,80 : POKE 26697,80 (Enter)
Now everything that would normally go to the screen will go to the printer. Drawback #1- No program line or immediate command can be entered that is longer than 32 characters (the length of the printer buffer). Longer program lines can be loaded from tape, before or after the Poke's, or typed in before the Poke's. Drawback #2- with an immediate command like: PRINT 2+2 (Enter) the answer (4) will overwrite the "P" in the word PRINT in the printer buffer before it is sent to the printer...so type in: PRINT 2+2 and then hold down the space bar to fill the printer buffer. When the printer starts to print, press enter and the answer (4) will print on the next line.

Yours Truly,

P. Aylesworth
Bradford, Ontario
Canada

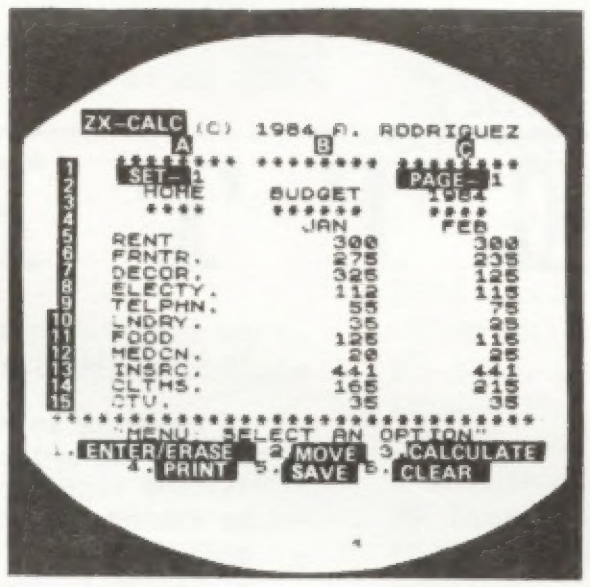
POWERFUL AND INEXPENSIVE BUSINESS SOFTWARE FOR ZX81, T/S1000 and T/S1500 COMPUTERS

ZX-TEXT

A word processor is to a computer user what a typewriter is to a typist, except that the former has more advantages than the latter. ZX-Text can operate in 16-64K RAM providing from 1300 to 6500 words per document. It features 6 different options: write, read, edit, print, save and clear text. Text is written on a per-line basis with quick speed and with horizontal back-space and delete capabilities being available. You can also access the editor directly from write mode and vice-versa. Text can be proof-read on a per-line basis allowing for enough time to determine if any editing is needed. The text editor allows a line of text to be deleted, inserted, replaced and listed for editing. You may also change a word or expression within a line, stop or start text while it is scrolling up the screen, begin reading text from the first line of the file, re-enter write mode from the editor, return to the main-menu or create a window so that you can read-edit two files simultaneously. The print option takes text displayed in 30-column format on the screen and outputs to either the ZX/TS printer. (With Memotech's Centronics Parallel Interface 80-column and lower/higher - case output is possible.) Files may be saved on tape cassette with the use of one single command, or by the same token they can be erased from memory / storage so that the full capacity of the program can be used for other purposes such as composing letters, reports, articles, memos, standard forms, instructions, ads, graphs, telephone directory, lists of customers, members, friends...etc. Also copies of files are always less expensive and easier to run than using a photocopier. Other advantages are savings in time, paper, ink, correcting mistakes and adding afterthoughts more efficiently than doing them through either handwriting or using a typewriter.

$16.95

ZX-CALC



An electronic spreadsheet calculator is the fundamental basic tool for summarising, reporting and analyzing in matrix form any accounting, mathematical or scientific manipulation of numbers. ZX-Calc operates in 32-64K RAM and affords a maximum of 3360 characters / spreadsheet. The entire matrix consists of 15 columns (letters A-O) and 30 rows (numbers 1-30) with 8 characters / cell. Unlike other popular ESCs, ZX-Calc uses in calculations and within cells all 14 math functions on the ZX-81 / TS1000. It offers a unique *SUM function that totals one or more rows / columns simultaneously. Parenthesis can be used within equations. There is no fixed limit on how many equations may be entered. Formulas may be stored in all 420 cells of the spreadsheet. The display affords 15 rows / colums. Loading of data into more than one cell can occur across / down one or more row / column simultaneously. With vertical windowing you can arrange a set of columns in any order, or practice using fixed-variable-alignment display formats. The menu offers 6 options: enter / erase, move, calculate, print, save and clear the spreadsheet. Enter / erase allows the entering, deletion or data alignment within a cell through the use of a mobile cursor. With the move option you may move around the entire sreadsheet to access any row, column or cell. The calculate option allows you to enter labels, values or formulas into a cell or write and enter equations that will act upon the data already within the spreadsheet. You can also enter bar graphs into a cell in this option. Absolute / relative replication, down / across a column / row, is also allowed by this option. Also this option allows the automatic calculation of the entire spreadsheet with one single command. Print allows you to output to either the ZX / TS printer the entire spreadsheet by column-sets and row-pages through use of the COPY command. The entire spreadsheet may be saved on cassette tape or you may clear all data from it or erase the program from RAM entirely. The most salient advantage provided by an ESC over specifically vertical applications software is that an ESC provides a reusable framework with which you can compose any specific financial model rather than just be limited to only one statically fixed format for storing, displaying and manipulating numerical data.

$16.95

ZX-CALENDAR

Time management is an important aspect of any serious business and personal agenda. Planning how to spend our time leaves us better prepared before and while we are spending it and we remain better organized after we finish spending it. ZX-Calendar operates in 16-64K RAM affording 25 appointments in 16K, 100 in 32K or 180 in 48K and 64K. Each appointment record holds a maximum of 220 characters. The main menu includes enter, search/check/sort, change, save, clear and print any and all appointments made on a specific date or with any party. Output to either the ZX/TS printer is permissible. This program will permit you to remember to do something or to be somewhere important by cataloging your answers to six questions that you must account for in order not to waste time when it is scarce: when, with whom, at what time, for how long, where and what are you going to discuss and conclude when you get together with someone else? The program lets you permanently originate, record, classify, search, sort, calculate, modify, summarize, obtain a written report and store your answers to the preceding questions so that you will not forget what you decide to do with your time. This program identifies your time according to when you are going to spend it and with whom you are going to share it. Through these forms of labeling appointments you are able to verify or modify how your time is budgeted without wasting ink, paper or more time trying to remember what you said to yourself or what someone else said to you or where you placed certain written messages that you now can't find. With this program you will know where you can find exactly what you need to know about where you want to and have to be, or where you have been, before you get and after you got there. Thus, ZX-Calendar will let you plan your time so that you will never have to worry about what is ahead or what came before, for you will always know, by using it, to never be caught astray by any time-frame.

$16.95

$3.00 SHIPPING AND HANDLING / PROGRAM

A.F.R. SOFTWARE - 1605 Pennsylvania Avenue, No. 204 - Miami Beach, Florida 33139

DEALER INQUIRIES WELCOME **(305) 531-6464** **FLORIDIANS ADD SALES TAX**

Goyette's "Ski" and "Cavern" A Hit

Dear Tim,

I found "SKI" (TDM-July/August 86) by Charles E. Goyette, to be a fantastic game. The only problem was that the person with the highest score for a game was not always listed as being the winner. Changing H$ to S$ in line 500 seems to correct this."

Sincerely,

Kenneth Fracchia
Buffalo, NY

Dear Time Designs,

I have enjoyed both "CAVERN" and "SKI" by Charles E. Goyette. They both act and react faster than my fingers can manipulate the keys. I did however, make a slight change in "SKI". I changed the trees that look like "bugs" to trees that look like trees with:

```
9000 DATA 1,128,1,128,3,192,3,19
2,7,224,7,224,1,128,1,128
```

This makes a nice pine tree with the addition of color, "INK 4", in line 1100.
For what it's worth.

Sincerely,

Richard B. McMahill
Washington, DC

Mathematics

Dear Sir,

Readers of TDM might be interested in the following equalities produced by my 2068. Other such relationships can be obtained by use of the program shown on page 232 of Laurie Buxton's book, "Mathematics For Everyone".

```
   PI - 103993/33102 = 0
EXP 1 -  49171/18089 = 0
SQR 2 -  66922/47321 = 0
SQR 3 -  70226/40545 = 0
SQR 10 - 168717/53353 = 0
  .125 - 1/8 = 5.8207661E-11
1.3 - 13/10 = 4.6566129E-10
```

Sincerely yours,
Howard R. Wilkerson

Greenville, SC

Request for LARKEN Help, etc.

Dear Sir,

I am writing in the hopes that you may be able to help me with a few questions. First of all I understand that there is a version of Prologue available for the Spectrum. I have searched all present and back issues of ZX Computing, Your Sinclair and Sinclair User that I own and could find no mention of it. I am hoping that you or one of your readers might know of the program I am talking about.

Secondly, I am hoping that someone might be able to help me with some conversion problems. I am trying to convert some of the other languages for the TS 2068 and the Spectrum onto the LARKEN disc drive system. I have Abersoft FORTH, Hi-Soft C, Hi-Soft Pascal and YS Mega-basic which I would like to make full compatable with the Larken system. So far, I have been able to put the main Basic loaders and machine code onto disc, but I do not know how to convert Save-Load routines within each language to save and load from disc. Perhaps someone has already solved the problem and could offer me some help. I thank you for your time and trouble.

Sincerely,

David Sölly
OTSUG Librarian
Ottawa, Ontario

Editor: I have found no reference to a "Prologue" program for the Spectrum, but some other "Speccy-phile" may provide the answer. As for your software conversion problems with the LARKEN disc drive interface, it sounds like you may have an earlier version of the LARKEN DOS. I have been in contact with Larry Kenny (a.k.a.; Larken Electronics) and he mentioned that a new 2068 DOS is available on disc that has improved LOAD/SAVE commands (supporting Arrays, Basic Code), and also FORMAT, CAT, ERASE and OPEN# commands. Larry also mentioned that he will begin work on offering the DOS (which is Spectrum compatible) on a cartridge, therefore using no computer RAM. Hold on to your seat on this one...it will have the NMI save feature. This will allow you to do "snap shot" saves like that on the John Oliger Disc Interface. I would suggest that anyone requiring information on new LARKEN improvements write to: Larken Electronics, RR#2 Navan, Ontario, Canada K4B-1H9.

"Pigskin Picks"

Dear Tim,

I am sending my check for another enjoyable year of TDM...the only real connection I have with the TS world, and I always look forward to receiving my TDM. I have enclosed a small football prediction program that I wrote. The program will average about 65% correct over the whole season.

I am also wanting to start a Users Group in the Bee County Area. If I can start one, it will be called "Bee County Timex Sinclair Users Group"- B.C.T.S., and if it is possible I would like to give the members that don't have computers a ZX81 or TS 1000 for joining. I would like to hear from other TS users for some input on this subject.

Sincerely,

Tom Proffitt
706 Morales St.
Beeville, TX 78102

Editor: "Pigskin Picks" was fun, but my team looked just as bleak as before I keyed in your program (better luck for me next year, I guess). Hope you get a users group off the ground. A free computer offer is hard to beat.

```
  1 REM "BY:TOM PROFFITT
      DATE:1982
    PLACE:BEEVILLE,TEXAS
  2 REM "Pigskin Picks" can als
o be used for Basketball. It wor
ks best after the fourth game. "
Not to be used for gambling if y
ou want to keep your money!"
  3 BORDER 1: PAPER 1: INK 7: C
LS : POKE 23609,70
 10 PRINT TAB 8;"PIGSKIN PICKS"
 11 PRINT "********************
***********"
 50 INPUT "enter 1st. team ";a$
 55 REM ENTER OFFENSE-POINTS
140 INPUT "POINTS-FOR ";c
150 REM ENTER DEFENSE-POINTS
160 INPUT "POINTS-AGAINST ";d
170 INPUT "enter games played "
;e
200 INPUT "enter 2nd.team ";f$
255 REM ENTER OFFENSE-POINTS
290 INPUT "POINTS-FOR ";i
300 REM ENTER DEFENSE-POINTS
320 INPUT "POINTS-AGAINST ";j
350 INPUT "enter games played "
;k
460 LET t=c/e
470 LET u=d/e
480 LET r=i/k
490 LET s=j/k
500 LPRINT a$;" ";INT (t+s)/2
510 LPRINT f$;" ";INT (u+r)/2
520 LPRINT
530 LPRINT
540 GO TO 10
```

sinclair News Network

New SPECTRUM Off To Giant Start
American Travelers Abroad Report on PC Show

American Timex Sinclair distributors Rob and Debbie Curry of Curry Computer and John Warburton of Sunset Electronics attended the annual Personal Computer Show in Olympia, Great Britain, the first weekend of September. The well-attended showing featured among other things, the premier of the Amstrad/Sinclair Spectrum 128k+2. Many thanks go to Mr. Warburton who thoughtfully picked up an extra brochure, which is pictured to the right, for Time Designs readers. The new Spectrum which replaces the previous 128k computer released six months ago by Sir Clive, offers both a professional full-travel keyboard and an integrated cassette recorder. It also has on-board twin joystick ports that use the Sinclair Interface 2 protocols (non Atari-type). Gone is the traditional black Sinclair look, for a new grey color.

It was curious that Commodore for the most part was absent at the show, while both Amstrad and Atari had huge displays. The Atari section featured many after-market companies, but all were integrated into the main Atari section with corresponding displays and decor... now that's company support! Meanwhile, Amstrad launched the new PC1512, an inexpensive IBM PC clone that is already receiving extremely rave reviews from the press. Watch for this one, it is rumored that it is coming to the U.S.

There were many software companies in attendance including an outlandish display by BEYOND, which replicated the bridge of the star ship Enterprise...a gimmick to announce their coming program, "Star Trek". Their were many other Spectrum related booths, and even some for the seemingly ill-fated Sinclair QL, such as the London-based support group, Quanta.

The Curry's stated that software and hardware "deals" struck at the PC Show, will greatly benefit U.S. Sinclair consumers in the coming months.

"All The News Fit To Print"

TS COMPUTERFEST II Plans Aired

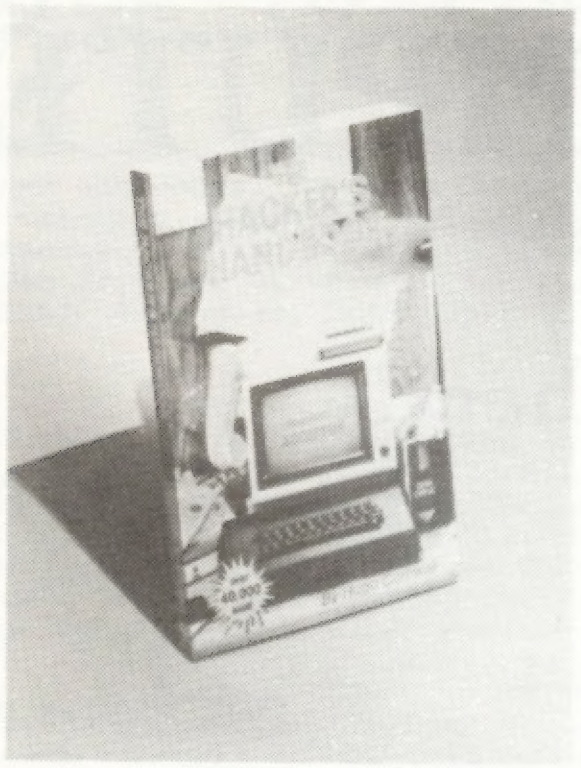
While May is months away, plans and groundwork for the Second Annual Mid West TS Computerfest continue. The "main event" this time will be held in Indianapolis, Indiana, on May 2nd and 3rd. It is being planned and hosted by nearly all of the representatives of the highly successful TS Computerfest held in Cincinnati last year, including Chairman, Frank Davis of Peru, Indiana.

Time Designs has been in contact with many of the dealers who attended the first show, and the overwhelming response has been "we'll be there again!". In fact several dealers who were unable to attend last year are definetly coming this time. Most preliminary figures estimate that the Indianapolis Computerfest will have double the attendance this time around, with perhaps as many as a thousand, now that the word is getting out.

Interested parties can write to Mr. Davis at: 513 East Main Street, Peru, IN 46970, for further details. Be sure and plan now to leave the first weekend in May open...you won't want to miss the Timex Sinclair "event of the year"!

ARCTAN COMPUTER VENTURES or Northampton, England, is an excellent source of support for the ZX81 or TS 1000 computers. The part software company and ZX81 magazine publishers have a five page brochure available. Arctan Computer Ventures (or A.C.V.), offers over a dozen different software titles, many of which are games (but also some utilities...like a Z80 Disassembler). The ARCTAN ZX81 Users Club has now published five exclusive magazines for ZX81 users. For complete information and prices, write to: A.C.V., 1 Foxwell Square, Southfields, Northhampton NN3 5AT, England.

Many months ago, we reported on the E. Arthur Brown Company of Alexandria, Minnesota, purchasing the exclusive U.S. publishing rights to England's popular computer telecommunications book, THE HACKER'S HANDBOOK. Now, Eben Brown (of E. Arthur Brown) reported to Time Designs, that the book is in it's second printing here. Hugo Cornwall, the author of the hacker's guidebook, made a scheduled appearance in San Francisco, California for a lecture at the "Hacker's 2.0 Conference", on the 25 and 26th of October. Mr. Cornwall is a noted international expert on modem "hacking". For information and prices on "The Hackers Handbook", write to E. Arthur

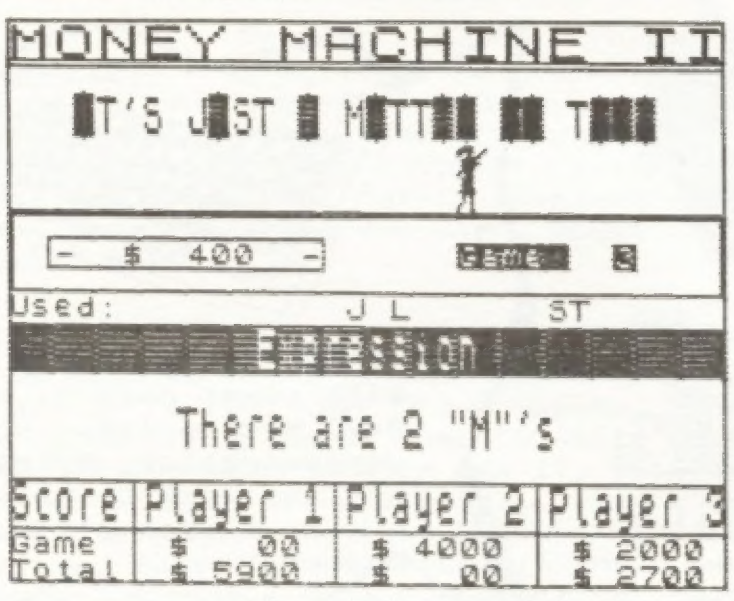


The Hacker's Handbook

Brown at: 3404 Pawnee Dr., Alexandria, MN 56308, or call (612) 762-8847.

DUNGEON OF YMIR Version Three is here. The all new high resolution maze game is available now for the ZX81 or TS 1000 that has both a 16k RAM pack and an 8k CMOS (static) RAM board such as the popular "Hunter NVM" board. Incidentally, if you have a copy of "Thrust" by the Weymil Corp., you are already set up to run Dungeon Of Ymir V3. Further details on this mega-game and other fine products for your ZX/TS, write to: Fred Nachbaur [Silicon Mountain Computers], C-12, Mtn. Stn. Group Box, Nelson, B.C. V1L 5P1.

Improvements on the "tried and true" appears to be the trend this month. In the May/June 86 issue of TDM, a program called "Money Machine" was mentioned for those that like word/thinking type games. We said that it resembled the TV game show "Wheel of Fortune". Now, the author has taken the program one more step...and we can now say that MONEY MACHINE II is a Wheel of Fortune clone. This should sell a lot of copies, as the TV show has gained a tremendous following. A lot of detail has been incorporated in this 2068 program, including a Vanna White ("Banna Brite" in the program, to protect author Herb Bowers from any legal implications) sprite that turns the letters. Play is conducted as in the show and up to three players can participate. Libraries of additional puzzles will be released periodically, but the 250 that come with the program should keep you and your friends up all night playing this one. Very good graphics and sound. Price is $15 from ABBA Soft, 2588 Woodshire Cir., Chesapeake, VA 23323.



Banna Brite turns the letters.

We've been impressed with all of the new stuff coming out of RMG Enterprises (1419 1/2 7th St., Oregon City, OR 97045) these days. New software titles include SOUNDESIGN (a utility for easy development of sound effects in your 2068 programs) and TRACER (a machine code utility for the 2068 that was inspired by a feature on the Sinclair QL. Interrupts allow the user to witness the actual execution of BASIC programs, as program lines are simultaneously displayed.) RMG also has excellent prices on disk drives, cases and power supplies, and many other items for the computer hobbyist. A new catalog is available for $2 (your $2 is deducted from your first order...so actually you pay nothing for the their catalog). Write for a copy.

Pete Fischer and Steve Ishii have put together the TS GUIDE TO TELECOMMUNICATIONS, which lists BBS phone numbers, hardware and software, and many useful tips. To obtain your own copy, write to P.O. Box 2002, Tempe, AZ 85281.

Have you blown your 2068's SCLD chip? Symptoms include video display problems, excessive LOAD and SAVE troubles, problems with internal clock timing and keyboard decoding, and others, you may have a faulty SCLD that needs to be replaced. You could send your 2068 in for repairs, but you can now (with some difficulty) replace your own. You may, or may not be aware that the SCLD is the only chip in your computer that isn't available commercially. Through the efforts of the Capitol Area Timex/Sinclair Users Group (P.O. Box 725, Bladensburg, MD 20710), which bought a large supply of them from the Timex computer factory in Portugal, you can now obtain these custom chips. C.A.T.S is offering them for $20 each ppd.

Knighted Computers, 707 Highland St., Fulton, NY 13069 (phone 315-593-8219) has obtained the U.S. rights to market TOMAHAWK, a combination helicopter flight simulation program and arcade game. Knighted has converted this popular Spectrum program to the stock Timex Sinclair 2068. The helicopter is a U.S. Army AH-64A APACHE, and features a 3D cockpit/window display, and use of both 2068 joystick ports! Price: $16.95.

TOMAHAWK has landed on U.S. shores.

A new ROM resident Monitor/Disassembler is now available for the Sinclair QL called ROMON. This comes from Meta Media Productions, 726 West 17th, Vancouver, B.C., Canada V5Z 1T9. ROMON 1.21 sports a host of features not usually found in a monitor. These include the display of SuperBASIC Functions & Procedures currently resident, the display of Jobs resident including the starting address and length of the job, the display of the major system variables and SuperBASIC variables, and more, in addition to the usual monitor functions of memory display and modification, register display, ect. ROMON is supplied on a ROMcard for the QL ROM port. Less than 1k of RAM is used for the storage of Monitor Variables. Write for pricing and further information.

Zebra Systems Inc., has just purchased the entire remaining stock of the popular SOFTSYNC line of 2068 software including the ZEUS ASSEMBLER, ZEUS MONITOR/DISASSEMBLER, Personal Accountant and several games. Zebra is now selling these commercial quality programs at a special price in time for the Holidays (stuff your stockings with these!). Consult their catalog or ads for further details, or write to: 78-06 Jamaica Ave., Woodhaven, NY 11421.

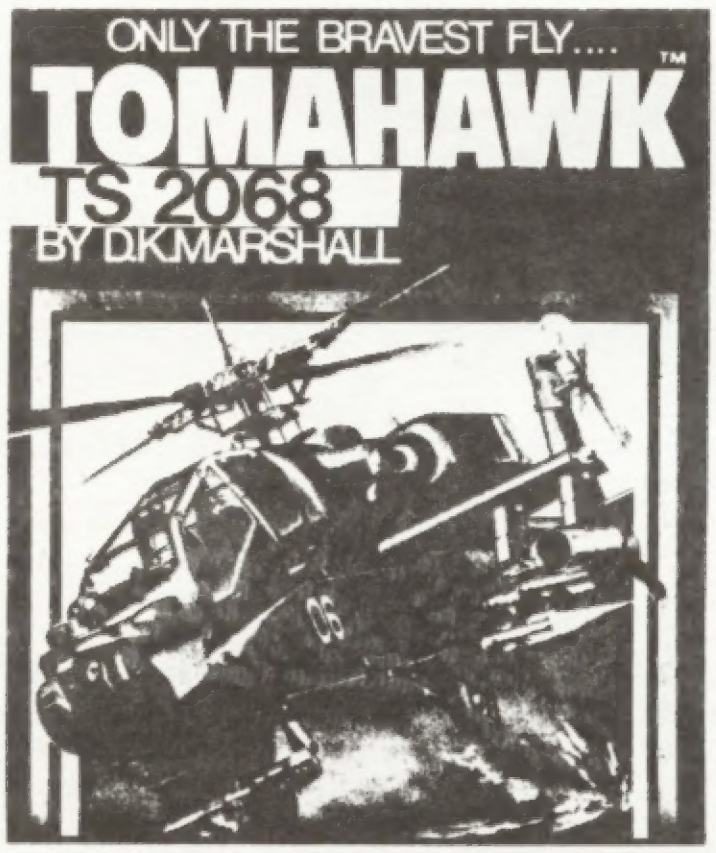
KNIGHTED COMPUTERS

707 Highland Street FULTON, NY 13069

(315) 593-8219

Software & Peripherals

TS 2068 ®



TOMAHAWK

TOMAHAWK is a real-time flight simulation based upon the US ARMY AH-64A APACHE Advanced Atack Helicopter - the meanest, deadliest combat helicopter ever to rule the skies! Its specialized job is to hunt tanks and destroy anything that gets in its way. The Apache was built specifically to fight and survive, night and day, in the thick of the battlefield.

Flying a real helicopter is a demanding task, requiring training and practice - particularly ground attack. TOMAHAWK gives you this challenge. Climb into your cockpit and prepart for take-off....

F E A T U R E S : Spectacular 3D real world display *Fully aerobatic (within limits of the real helicopter) *Ground attack & air-to-air interception *Over 7000 ground features *Day/Night vision systems *Cloudy conditions, crosswinds & turbulence *Doppler navigation & target tracking system *Laser guided missiles, plus rockets & 30mm chain gun *Selection of training and combat missions *Impressive sound effects *Pilot ratings - Trainee to Ace *Uses BOTH TS2068 joystick ports!

ITEM # 1212 $16.95

Sinclair QL®

MICRODRIVE CARTRIDGES
(pk of 4) Item #1158 $ 11.95

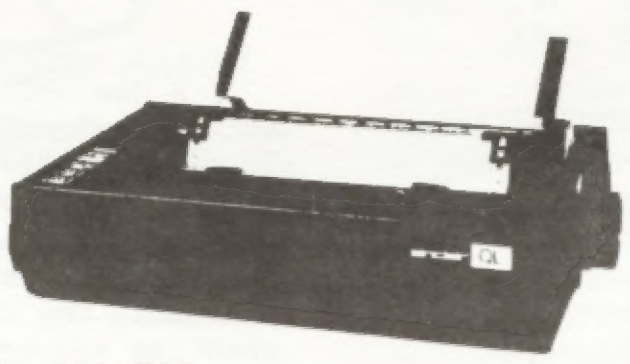
TRANSFORM BOX - HOLDS 20 CARTGS.
Item #1205 $ 9.95

512K RAM EXPANSION

This 512K card increases the QL's memory to 640K of Random Access Memory. With this memory expansion you can take full advantage of your QL. Our memory board is equipped with thru-porting so that you will still be able to connect a disk drive interface. This is another high quality product from KNIGHTED COMPUTERS.

Item #1069 $199.95

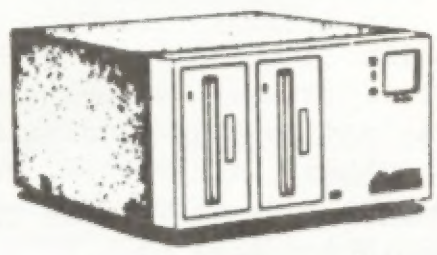
HARDWARE FOR YOUR QL COMPUTER



QL PRINTER
80 CPS, 9 PIN DOT MATRIX, AND COMES WITH LQ MODE (LETTER QUALITY) AND CABLE TO HOOK UP DIRECTLY TO YOUR QL SERIAL PORT.
Item # 1198 $199.00
QL PRINTER RIBBON
Item # 1180 $ 11.95

BEST BUY

DUAL 3" DISK DRIVES



O N L Y $ 2 4 9 . 9 5 ! !

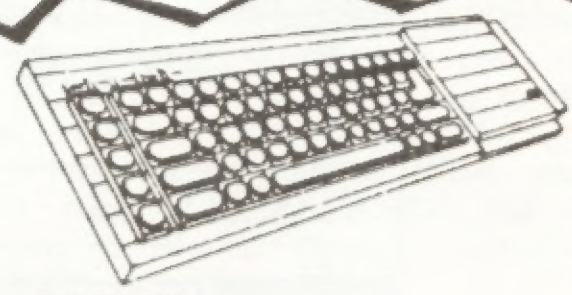
These top quality dual disk drive units have their own power supply and fan built-in, and very attractively encased. Now, cut your data storage costs by more than half. These drives are single sided, double density drives and format out at 180K per side - with just two disks, you'll have data storage capabilities of 720K - and the convienience of having two drives available at your command. INCLUDES: CABLE AND CONNECTORS AND QL DISK DRIVE INTERFACE, AND ALL FOUR OF THE QL PSION PROGRAMS ARE ALREADY PUT ONTO TWO DISKS FOR YOU.

ITEM# 1210 $249.95

3" DISKS (for above)

Hard plastic encased top grade 3" disks. Nice sliding metalic disk protector, and write protect locks for each side.

ITEM# 1211 $ 4.50
(BOX OF 10) ITEM# 1212 $40.25



QL COMPUTER -
With power supply, manual, dual microdrives, Data-base, Wordprocessor, Spreadsheet, and Business Graphics programs
Item #1140 $209.95

VISA

MasterCard

GUIDE TO TIME DESIGNS BACK ISSUES

COMPILED BY PAUL BINGHAM

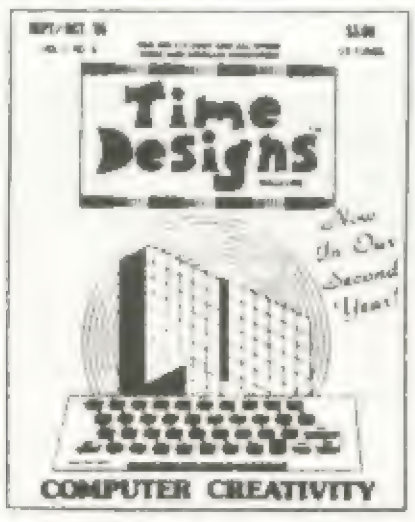
Time Designs Nov/Dec '84 Vol 1, No. 1
TS2050 Modem announced, Spectrum Emulator announced, U.S.A. QL released, 2068 Tech Manual released, Recoton Cassettes, "Rise & Fall if Timex Computer Corp.", 2068 Software Directory, Portable TS project, BASIC languages, 2068 Music I, 2068 Tasword II review

Time Designs Jan/Feb '85 Vol 1, No. 2
MT Sprit Printer (TS2080), Comparing Data Base Services, TS Modems, 2068 MTERM II review, TS1000 & 2068 Screen storage prgrm, 2068 "Ant Attack" review, ROMSWITCH review, Static Discharge Bracelet, 2068 BREAKing and SAVE, 2068 "Great Game/Graphics Show" review, 2068 "Compass" compiler review, 2068 "Straits of Hormuz"&"Exec Clue" review

Time Designs Mar/Apr '85 Vol 1, No. 3
Timex & Dave Higginbottom, TS User Group Directory, 2068 Text Entry/recall in M.C., TS1000 DATA/READ, 2068 House Payment prgrm, TS1000 "Basload" & "Super-tape" reviews, Guidelines on Over-seas Ordering, 2068 Z-LINK I/F review, 2068 OMNI-EMU review, Spctrm "White Lightning", "Lords of Midnight" & "BRAxx BLUFF" reviews

Time Designs May/Jun '85 Vol 1, No. 4
User Group Update, Oliger/Kingsley Disk I/F announced, TS2068 returning via Portugal, TS1000 "Intro to Computer Control", 2068 M.C. joystick routine, 2068 Graphics Problem, TV to Monitor kit, Tasword II Tips, 2068 Pro/File review, 2068 "Bill's Coupon Magic" & "Dealer's Den" reviews, Damco's Spectrum Wafadrive review

Time Designs Jul/Aug '85 Vol 1, No. 5
Timex Network, Softaid hunger relief, Users Group Update, 2068 Graphics, 2068 Tic-Tac-Toe prgrm, TS 1000/2068 Alphabetizer prgrm, TS1000 Set RAMTOP, TS1000 8K upgrade & Bank Switching reviews, Aerco FD-68 2068 Disk I/F review, 2068 Zebra-Talker review, 2068 "Night Gunner"&"Grade Book" & Spectrum "Deus ex Machina","Pajamarama" & "Sherlock" reviews

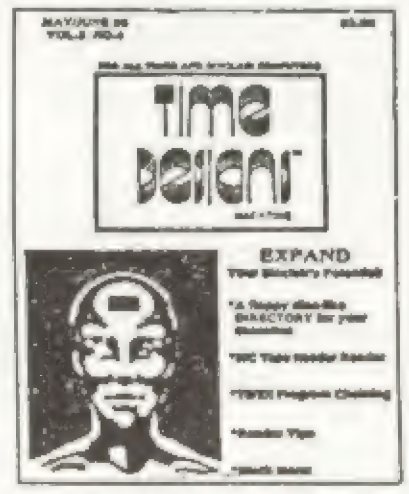


Time Designs Sept/Oct '85 Vol 1, No. 6
Portugal 2068 Update, TS1000/2068 Pablo Pixel-o HIRES Graphics prgrm, TS1000 Adventure in the RAM Jungle, TS1500 Monitor Adaptor, 2068 Zebra Graphics Tablet review, 2068 Zip Compiler review, Book review "Minute Manual for Dot Matrix Printers", User Group Update, Damco Spectrum Emulator, Spectrum "American Football" review

Time Designs Nov/Dec '85 Vol 2, No. 1
U.S. QL reduced to $299, 128K Spectrum released, Sinclair TV review, User Group Update, 2068 Gazer's Guide to Halley's Comet, 2068 "Shuttle Designer" review, TS1000 Adventure in RAM II, TS1000 Interest prgrm, FD-68 Disk I/F review II, Zebra 2068 Disk Drive review, 2068 Obsticle Run prgrm, 2068 ZPRINT-80 review, 2068 VIEWord/MAIList/FORMail review, Spectrum D'KTronics Speech Synthesizer review

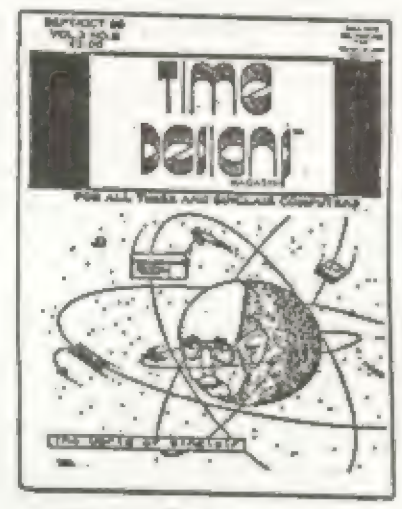
Time Designs Jan/Feb '86 Vol 2, No. 2
"Why The QL?", TS1000 Adventures in RAM III, TS1000 Chroma-Soft review, 2068 Shell Game prgrm, Linear Programming for TS user, 2068 "Turbos" engine prgrm, Radio Shack Mouse & Zebra 2068 Graphics Tablet, 2068 "lollipops" prgrm, Gamesmate fix, Joystick Wrap Around prgrm, 2068 Labelmaker prgrm, Portuguese Disk Drive for 2068, 2068 Machine Code Tutor review, 2068 OS-64 review, 2068 Rainbow Plus emulator review, Spectrum "Astronomer" review

Time Designs Mar/Apr '86 Vol 2, No. 3
Programming in QL SuperBASIC, QL "GraphiQL" review, "Of Strings and Things" TS1000, Z80 Machine Code I, 2068 Tasprint & Aerco printer I/F, 2068 Phone # prgrm, Convert WC2050 Modem to RS-232 I/F, How to connect with BBS, 2068 Burglar Alarm, HI Res/64 col utility 2068, Oliger 2068 Disk I/F review, 2068 "ARTIST" review, Top-10 2068/Spectrum prgrms, Mini amp for Spectrum SW



Time Designs May/June '86 Vol 2, No. 4
Five TS1000 tips, Printer tips, improve Zebra Graphics Tablet, NEWS: Amstrad Buys Sinclair, TS Computerfest Report, User Group Update, 640K QL upgrade, QL "Qspell" review, TS1000 Consentration game, TS1000 prgrm chaining I, Z80 Mchn Code II, 2068 Cassette Directory prgrm, 2068 header reading w/ Mchn Code, 2068 Cavern game, Moving an AROS cartridge to FD-68, Adding a Spectrum/2068 joystick, More on Oliger Disk I/F, 2068 Pro/File Extentions review, 2068 "Address Book" review

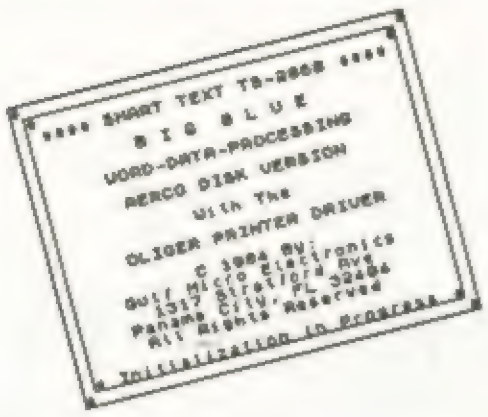
Time Designs July/Aug '86 Vol 2, No. 5
Sinclair Micro Update, Meet the QL Clones, Time Designs Acquires S.U.M., QL Game reviews: (Matchpoint, Chess, War in the East, Wanderer, Squadrons & Hyperdrive), QL "Cosmos" review, TS1000 External Keyboard Buffer, TS1000 Digital Clock, TS1000 prgrm chaining II, Z80 Mchn Code III, 2068 Bankswitching "Missing 253", Embellishing 2068 MTERM II, 2068 Ultra-Easy Designer Graphics, 2068 "Poly-scroll" prgrm, 2068 Video file prgrm, Datagen 2068 DATA statement prgrm, 2068 Ski game, 2068 Tasword Word Count addition, 2068 "MacIntosh" Menu, 2068 Sound Synthesizer review, Spectrum "Saboteur"review, 2068 "Colonize the Universe" review

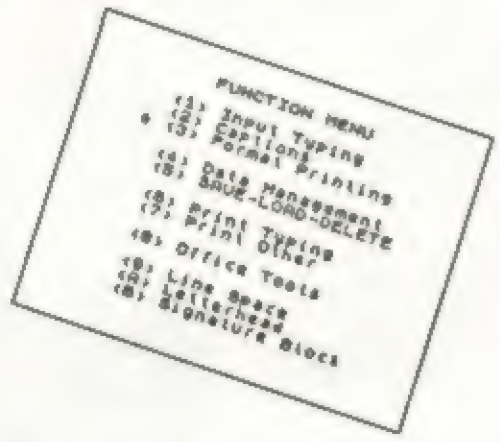


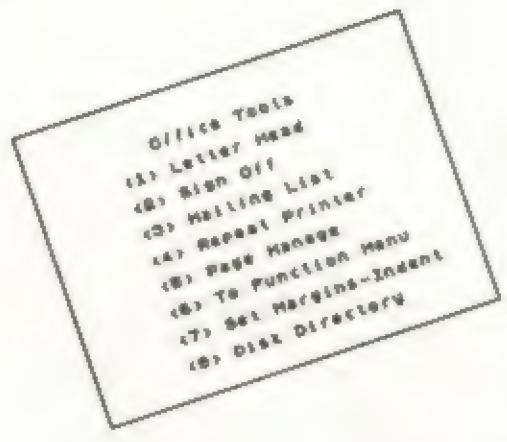
Time Designs Sept/Oct Vol 2, No.6
Sir Clive's Confessions, User Group Update, "Light Show 2000" 2068 prgrm, 2068 Pixel Sketch & Graphics Editor review, 2068 Timachine Compiler review, Larken TS1000 Disk I/F review, TS1000 ZX-CALC + R.F.R.G. review, 2068 Bank switching: more about missing 253, 2068 3-D Tic-Tac-Toe prgrm, QL Peintre review, QL,s U.S.A. future, QL Graphic/CAD Systems review, Z80 Mchn Code IV, TS1000 Prgrm Chaining III, Upgrading TS1016 RAM to 64K

SOFTWARE IN REVIEW

SMART TEXT TS-2068







A warm grin began to pass over my face late last night. In my mind's eye I could see a little light bulb beginning to glow dimly in the cartoon balloon above my head.

I was curled up with a good book, uh manual. After my second reading of this thirty page treatise and three sessions with the two hour electronic introduction to the program, the concepts by which it functioned were finally beginning to fall into place. I was becoming a cursor instead of a curser.

"SMART TEXT TS-2068" is Bill Jones' effort to create an AppleWorks environment for your TS-2068. The package includes a text editor and manager integrated with a small mailing list routine. Other convenient features are printer formatting for a variety of utilitarian purposes and in a variety of type styles. The $39.95 price tag makes SMART TEXT one of the more expensive programs available for the 2068. The programming and memory saving tricks alone seem to justify the cost. In addition to some valuable techniques in data management, you can also perform some useful tasks with SMART TEXT.

Of the programming tricks mentioned, three are significant. The first is the routine that manages text entry. The character code of the keystroke is checked for validity. This approach is preferable to use of the INKEY$ function. It allows, from BASIC, a typing speed of 100 words per minute, according to the manual. Read this as you would a mileage claim on a car window sticker.

SMART TEXT makes extensive use of memory saving techniques we learned in our TS 1000 days. Two old standards are employed by SMART TEXT to include within the software as many functions as possible while still retaining enough memory for a decent amount of text storage.

One of these is the use of letter variables to represent often used numbers, including program line numbers. "Pseudo Hex" is a term coined by Bill Jones to refer to his assigning of the variables oo, oa, ob, ect. to represent the numbers 0, 1, 2, ect. Another memory saver is the use of logical operators in long single program lines to replace the need for many lines to act on menu choices. Bill calls the technique "Dense Pack BASIC".

SMART TEXT functions are many, varied, and utilitarian. This software appears to have been developed by a "user", first for himself, and now for others. There are so many choices not offered in other word processors that it will take the new user a good number of hours to appreciate the alternatives presented by this text editor and manager.

The first and most basic function is typing. Like other word processors, SMART TEXT allows you to delete mistakes and to retype. You can also insert new text between already typed words. Entire blocks of text can be deleted and inserted. Additional text can be appended to the original, and the whole text file can be saved. Pretty standard stuff.

Unlike other word processors I have seen, SMART TEXT allows you to print what you have just typed without having to make a lot of decisions about how the printer should format the output. The text you've entered is quickly printed perfectly centered on the page in any print style but without any embedded printer commands.

The secret to this "smart typewriter" mode is the fact that you have already spent some time explaining to SMART TEXT the various commands your printer requires to enable and disable any special modes or pitches.

Program lines must be revised, within the guidelines of the Oliger protocol, to set up your printer's various capabilities. I altered the review copy to reflect the pica, elite, condensed, and proportional pitches available on my Prowriter as well as its bold print and double width modes.

SMART TEXT automatically calculates the maximum length of a text line in the selected pitch or mode. It asks you what length line you want to print. Then it calculates the correct margins and adjusts accordingly.

One of the reasons my hair is grey is the time I've spent calculating margins for center printing different print pitches and widths. Embedded commands that are counted in some word processors and not counted in others have driven me to considerable distraction. No longer.

In addition to printing text centered on the page, SMART TEXT provides the capabilities of center printing captions and letter heads, flush left printing of addresses and salutations for business-like letters, and automatic formatting and printing of the signature block of a letter.

Printing form letters to different people is accomplished by the integration of a small mailing list. Twenty-four records can be added, displayed, corrected, deleted, and saved.

Mailing labels or form feed envelopes can be printed. The mailing list is also used to "personalize" form letters with the first name of the recipient. A comma should be appended to the first name field, since the software does not include it.

In addition to letter formatting, SMART TEXT TS-2068 assists with the printing of manuscripts, documents other than letters. The document can be printed with justified or non-justified right margins, with the first line of each paragraph indented or with the entire paragraph in block form, and with entire paragraphs block indented with properly adjusted margins.

The key to the preparation of manuscripts is the concept of the paragraph. SMART TEXT is set up to store discrete paragraphs in separate elements of two string arrays. It can alternately be organized to store con-

tinuous text in three large strings. When one string is filled, the text automatically moves into the next consecutive string.

Discrete paragraphs are stored in the H$ and I$ arrays. The dimension of the elements in the arrays is user slected. The maximum is about 850 characters each. Ideally the text stored in these paragraphs has already been edited and does not need to be changed.

Continuous text is stored in the A, B, and C "banks". Any of these banks can be reviewed and edited via menu selections. The user can selectively copy a portion of one of these banks to another string (L$), called "The Paragraph".

This storage area can be altered or appended fore and aft. These editing functions can occur even if you are currently entering new text into the typing buffer (U$).

Prior to any editing action, any text that may be currently in the typing buffer is temporarily "set aside" into and S$ storage area. The text to be edited is then placed into the typing buffer for viewing or alteration. When the editing is completed, all the text is automatically restored to its former position.

When all your text has been edited and ready for printing, you have a veritable plethora of printing sequences from which to choose. The text may have been stored in up to twenty different positions. User alterable program lines determine which text is printed and in which order.

Repeat printing functions take care of the number of copies and the page formatting. Fifty-four lines are printed per page, the pages are automatically numbered, and form feeds are sent at the appropriate times. Go make yourself a cup of coffee.

My Aerco Disk version of SMART TEXT makes excellent use of those areas of DOS which have been implemented and works around those that have not. A lengthy tutorial is included.

The tutorial is filled with bells and whistles that tended to get on my nerves after a while. A list of the clever graphics and sounds is provided by the tutorial to serve a reference for the use of these techniques in your own programming.

The tutorial and the software are both tributes to the fact that the BASIC syntax checker of our Timex Sinclair computer will not forgive misspellings and grammatical errors except in "Print" statements. This untidiness detracts from the cosmetic appearance of the software, although it does not affect its usefulness.

SMART TEXT is available for cassette users in both 32 column and OS-64 versions. A&J Micro Drive, Aerco Disk, and Oliger Disk versions are also available. Aerco, A&J, Tasman, and Oliger printer interfaces are supported. Specify version.

SMART TEXT is $39.95 and is available from Bill Jones, Gulf Micro Electronics, 1317 Stratford Ave., Panama City, FL 32404. Bill welcomes your comments and questions. Call him after 6 pm local time at (904) 871-4513. You'll enjoy the experience.

--Duncan Teague

MUSICIAN ROYAL

MUSICIAN ROYAL is one of the most recent programs released for the 2068. Written by Dr. Oleg D. Jefimenko and sold by Electret Scientific Company, it proves to be one of the more comprehensive music programs available.

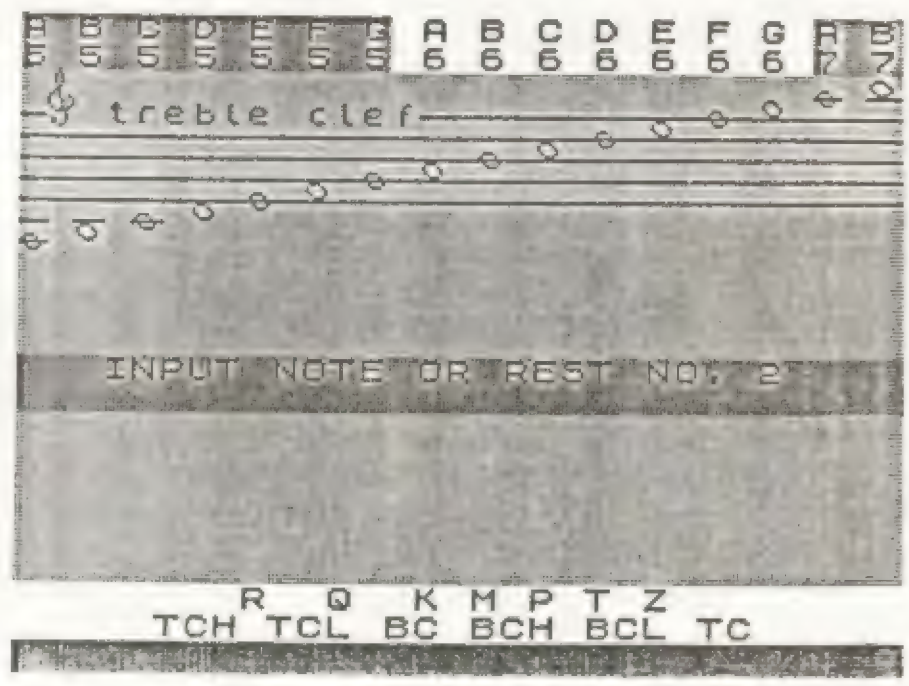
One of the most useful features of the program is the ability to transcribe already written music into the computer and have it play it back to you - using the BEEP command. Even though only one voice is available, the control over the parameters makes up for it.

The play options allow you to change the key in which the composition is played, the tempo can be changed as well as the order in which the song(s) can be played. The editing features allow you to change any possible errors.

The tape comes with three programs as well as a demo program with six compositions already transcribed and ready to play. The first program is the actual transcribing program where you are greeted by a screen that asks you for the name of your composition and the number of sharps or flats contained in your selection. You are then greeted with a musical staff with notes and their respective pitch (several octaves worth) graphically pictured on the screen. A prompt with several different menu choices are also displayed. You enter the notes one at a time adding the inflections (sharp, flat or natural) as needed. The treble clef is displayed but you can change it up or down an octave and also the same can be done with the bass clef.

Next you enter the value of the note for the time signature (whole, half, quarter note, ect.). At the end of each measure, you can enter a Q which is an aid when going back and editing. A duplicate function is also available when you have two groups of notes that are in the same order which really saves some typing. Entering Z allows you to correct your last entry.

With a printer (TS 2040) attached, the information is printed as it is entered so as to see where you are and to make it easy when looking for entry errors. Once the transcribing is completed, you have the option of



playing, SAVEing, LPRINTing, or editing. You can have it play as written, or in any order you wish, or continually repeat itself.

The manual is very well written and leads you through the program carefully. The second chapter is devoted to those with very little background in music. It gives you a crash course in music notation and what all "all them little symbols mean".

The second program on the tape is called MUSIC BOX and it allows you to take the music transcribed in MUSICIAN ROYAL and collect them. Each MUSIC BOX that you make can hold up to 8 compositions with up to 1500 notes (total) in the first seven and 1500 notes in the eighth composition. MUSIC BOX is easily filled by loading in data saved from MUSICIAN ROYAL. A table of contents helps you keep track of what is stored already.

MUSIC ALBUM is called the "ultimate program" for collecting and playing compositions transcribed in MUSICIAN ROYAL. It can hold up to 2000 notes total in 8 compositions. You have the most control over the tonality of each composition. You can have the slections play in any order, control the tempo, and control the duration of the pauses between compositions.

The programs all have certain safeguards built in, but are all easy to convert to mass storage such as disk drives. Large printer drivers are also easily added.

All in all, the program is very professionally done from the packaging to the 75 page manual which comes with it. I had no problem loading the tape and my order was delivered within three weeks of placing the order by letter.

Needless to say, I was disappointed that it only used the BEEP command. I remember that I was so excited when I saw the ad and ordered it thinking "Great! A decent looking SOUND program has finally been written for the 2068". At least I assumed it used SOUND. Because of this, I think the $20 price tag is a bit steep. It would be well worth it if it used SOUND with all of the features it contains. Hopefully Dr. Jefimenko will come out with a sequel using all four voices.

The program is available from Electret Scientific Company, PO Box 4132, Star City, WV 26505 for $20. If you would like a sample of what you can expect, (a nice courtesy) they will send you the DEMO ALBUM for $3 which will be subtracted from the $20 if you do decide to order the whole program.

--Joe Williamson

SPRITES 2068

Sprites, for those of you unfamiliar with them, are very simply User Defined Graphics (UDG's) which are capable of moving about the screen. Aside from movement, one large difference from regular UDG's is that sprites are normally larger than a single character space. A good example of a sprite, is the ghouls and goblins that appear on the screen of most arcade games.

SPRITES 2068 is a sprite development and handler package. Those of you that have the TDM Technical Manual will note that there is an appendix, number C-5, devoted to this subject. This program is none other than this same code.

SPRITES 2068 co-authors Tidwell and Ruegg have "debugged" the Timex sprite package code. This was no small task as I had tackled this myself and found many "bugs". They have also enhanced the original package with the addition of an automatic RAMTOP setter and additional screen attribute capabilities.

Having the Timex sprite service code is of no use however, if you cannot interface a program with it. Therein lies the true value in this package as Tidwell and Ruegg have prepared a comprehensive manual and a very nice Basic program to demonstrate the sprite packages' abilities. The user manual is clear and easy to understand. They have assumed you know nothing on the subject, yet, have not "talked down" to the advanced programmer.

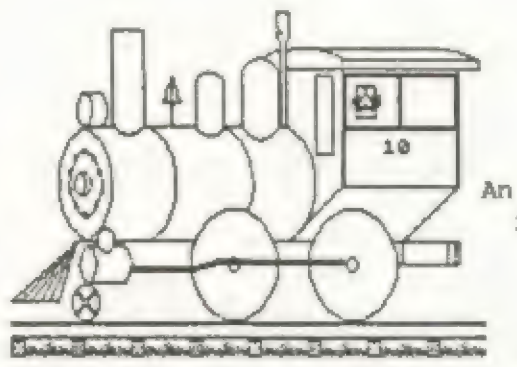
Authors, Tidwell and Ruegg have included in the manual, a section on machine code interfacing. This section is the poorest part of the manual, but if you can write machine code programs, you will not suffer for it. They have thoughtfully listed all of the variables and a memory map, and of course...you already have your own copy of the Technical Manual to go by.

And as if all of that were not enough, there is also a very nice UDG development tool included with the Basic demo program. This could be used alone to aid in the addition of UDG's to your programs. It allows the design of each UDG in enlarged format and then displays the UDG in normal size, as it would appear on your screen. It will also display a group of UDG's, 8 across by 8 down, to view your sprite (or a portion of it) as it will appear.

SPRITES 2068 will allow up to 256 sprites, each one up to 256 by 256 characters. In practice however, you will find the constraints of memory size will not allow for this. The invisible wall, RAMTOP, will not interfere with your use of sprites, as there is enough memory in the 2068 for most all the sprites you will want to use.

The smallest sprite possible is one character space (8 by 8 pixels), due to the use of the UDG's as designed by Timex. This means that your sprites will require some thought as you can only use two colors in each character space. Also, movement of the sprites can appear "blinky" if there is too much going on in your program.

There are vertical and horizontal screen scrolls in SPRITES 2068, however, they too use the character space



An animated sprite display from the demo program of SPRITES 2068.

as the smallest unit of measure. They can be combined to create a scroll in eight different directions. The scrolls, as well as the sprites should really be addressed on a pixel-by-pixel basis. However, it appears that Timex never intended us to have that kind of control from the "normal" video mode.

SPRITES 2068 is a crude sprite package BUT it is the ONLY one available for the Timex Sinclair 2068. If you are accustomed to the graphics abilities of other computers, such as Commodore or Atari machines, you might be a little disappointed. Keep in mind, however, that those computers were initially designed to play games and therefore, have sprite capabilities as part of their operating systems.

I found SPRITES 2068 to be an excellent "starter" package. If you want to include sprites in your own programs, you will find this utility very easy to use. I would suggest that you do follow the user manual's suggestion of programming in small blocks, as you must be very careful to maintain control over what is happening on the screen. This control is needed due to SPRITES 2068 use of the Attr-P system variables instead of Attr-T. With some careful planning, you will be amazed at the results you can accomplish.

Tidwell and Ruegg deserve a big hand for their thorough treatment of sprites. They have taken the Timex sprite routines and explained them to us in laymans terms.

Price for the SPRITES 2068 development package on cassette, complete with a comprehensive 34 page manual, and an educational (and entertaining) demo program, is $19 ppd. It is available directly from the authors (Vern Tidwell- 1303 Whitehead St., Key West, FL 33040 or Ron Ruegg- 37529 Perkins Road, Prairieville, LA 70769) and some Timex dealers handle it also.

--Syd Wyncoop

KALEIDOSCOPE

A Superior Version Of This "Old Standard"

For Your ZX81/TS 1000 And 1600

By Zack Xavier Haquer

Many "Kaleidoscope" programs have appeared over the years. These simple, but fascinating graphics displays have been adapted to virtually every computer ever built. The ZX81/TS family is no exception. Unfortunately, since the Z80 CPU in these machines is (effectively) clocked at only .5 mHZ in SLOW mode, the result is rather slow and BO-RING. Type in the program Listing #1 (BASIC prototype), and you'll see what I mean.

Now, let's take essentially the same program and write it in Z80 machine code. For a graphic demonstration (pun intended) of the speed and compactness of code ...start by entering a 1 REM line, followed by 172 X's or other character. (HINT: use FAST mode.) Now enter the rest of Listing #2 (machine code loader). RUN the program, and input the values given in Table #1 (decimal machine code). Go from left to right, top to bottom. Take your time, and proof each number before you enter it.

When you're done, your 1 REM line will look like sheer nonsense. Don't worry about that; just check it against the 1 REM in listing #3 to make sure it's the same. Enter lines 2-6 of Listing #3, overwriting the loader. Delete lines 7 and 8. SAVE to tape with RUN 5. The program will auto-run when the save is finished. WOW! Press BREAK when you're adequately hypnotized.

Line 2 contains the fill characters used for the display. Change this however you see fit; there is no limit on length. HINT: use symmetrical characters, like O, *, =, +, the grey squares, spaces, and their inverses.

This will run on 2k machines if you modify the BASIC portion as shown in listing #4 (2k changes). This is because this program relies on a fully padded-out display file.

If you're interested in studying the machine code, use a disassembler or HOT Z to take a look at the code. As mentioned, it pretty much follows the structure of the BASIC prototype, so you should have little trouble finding your way around. the code from 408Fh to 49B4h gets pseudo-random numbers in B and C. Next are two ways of implementing a modulo function. At 40B5, C is reduced mod 16 (exact power of 2), and at 40BD B is reduced mod 12 (not a power of two). The CALLs to 40F6, 40FB, 4100, 4105 and 410A correspond with the BASIC GOSUBs to 250, 300, 350, 400 and 1000, respectively. The routine at 410A is a "print at BC" routine which is MUCH faster than the comparable ROM call to 08F5 followed by RST 10h. It prints the character pointed to by CH_ADD at row B, column C. NOTE: it does NOT check for over-range.

The next time some smart-aleck ribs you about your "slow" ZX/TS, boot this program and watch his jaw sag. Isn't this fun?

LISTING 3: M/C KALEIDOSCOPE

```
   1 REM )NINKEY$(, RETURN
A487 -RND)?  GOSUB ?MRND=??LN =(
 FAST ??LN =(AT ?▗K"▗??▩K"9?F6MR
ND??ACS TACS TACS TACS T▗"£▩S=▩/
IF ?VAL LN PLOT RNDLN  INKEY$LN
=INKEY$AT VAL LN PLOT RNDLN ▌INK
EY$LN =INKEY$AT VAL LN CLS RNDLN
  INKEY$LN =INKEY$AT LN CLS RNDL
N ▌INKEY$LN =INKEY$LN ??S▩TAN Y(
▗?TAN Y(▩/ RAND Y£▩?TAN Y£▩/ RAN
D E£RND7VAL : ACS SACS ;ACS SACS
 ;ACS SACS ;=AT ?▗" ?= GOSUB ?-R
ND,ACS ▣?TAN
   2 REM ▀ ▒▒▒
   3 SLOW
   4 RAND USR 16514
   5 SAVE "KSCOPE"
   6 RUN
```

LISTING 1: BASIC Prototype

```
  1C▩LET F$="▀ ▒▩▒"
  20 POKE 16418,0
  30 LET POINT=0
  40 LET POINT=POINT+1
  50 IF POINT>LEN F$ THEN GOTO 3
0
  60 LET A$=F$(POINT)
  70 LET B=INT (RND*12)
  80 LET C=INT (RND*16)
  90 GOSUB 250
 100 GOSUB 350
 110 GOSUB 1000
 120 GOSUB 250
 130 GOSUB 400
 140 GOSUB 1000
 150 GOSUB 300
 160 GOSUB 350
 170 GOSUB 1000
 180 GOSUB 300
 190 GOSUB 400
 200 GOSUB 1000
 210 GOTO 40
 250 LET X=16+C
 260 RETURN
 300 LET X=16-C
 310 RETURN
 350 LET Y=12+B
 360 RETURN
 400 LET Y=12-B
 410 RETURN
1000 PRINT AT Y,X;A$
1010 RETURN
```

LISTING 2: Machine-code Loader

```
   1 REM XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
   2 FAST
   3 FOR A=16514 TO 16685
   4 SCROLL
   5 INPUT B
   6 POKE A,B
   7 PRINT A,PEEK A
   8 NEXT A
```

TABLE 1: Machine-code Decimal Data

```
 17: 51: 65: 19: 26:254:118: 40:
247:237: 83: 22: 64: 17: 75:  0:
237: 75: 50: 64:  3: 98:104:205:
  9: 19:229: 98:105:205:  9: 19:
193:124:129: 48:  1:  4:103:125:
144: 48:  1: 37:111: 43: 34: 50:
 64: 69: 76:203: 57:203: 57:203:
 57:203: 57:  6: 12:184: 56:  3:
144: 24:250: 71:197:205:246: 64:
205:  0: 65:205: 10: 65:193:197:
205:246: 64:205:  5: 65:205: 10:
 65:193:197:205:251: 64:205:  0:
 65:205: 10: 65:193:205:251: 64:
205:  5: 65:205: 10: 65:205: 70:
 15: 56:144:201: 62: 16:129: 79:
201: 62: 16:145: 24:249: 62: 12:
128: 71:201: 62: 12:144: 24:249:
 42: 12: 64: 35:197: 14:  0:203:
 56:203: 25:203: 56:203: 25:203:
 56:203: 25:  9:193:120:129:  6:
  0: 79:  9:237: 91: 22: 64: 26:
203:183:119:201:
```

LISTING 4: 2K Changes

```
  3 SLOW
  4 POKE 16418,0
  5 FOR A=0 TO 23
  6 PRINT AT A,31;" "
  7 NEXT A
  8 POKE 16418,2
  9 RAND USR 16514
 10 SAVE "KSCOPE2K"
 11 RUN
```

SPRITES 2068

FANTASTIC NEW PROGRAM! Exciting & Educational

You've heard of SPRITES. Explore the subject. Create Moving Displays and Games. Buy a copy for your TS-2068, they belong together.

FEATURES:
- SPRITE SERVICE UTILITY (2520 Bytes IMPROVED professionally written machine code)
- SPRITEDRAW Program (Great NEW program makes drawing a SPRITE both fun and easy)
- Operate from BASIC (Utilizes a Machine Code Interface. Twelve SPRITE Commands)
- Operate from Machine Language (VERY QUICK Screen Action! Bonus MCSPRITES program)
- Cassette Tape (Includes demonstrations and instructions. Menu driven. Fun to watch)
- Manual (34 pages written in "user friendly" style for all programmers)

Authors INTRODUCTORY SPECIAL Includes Postage
To ORDER: Send Check or Money Order $19.00 To

VERN TIDWELL
1303 Whitehead St.
Key West,FL 33040

OR

RON RUEGG
37529 Perkins Road
Prairieville,LA 70769

"I have seen SPRITES 2068. I have read it and I have tested it; and I cannot say enough good about it. If I had to sum it all up, I'd say that if you HAVE a TS-2068, you NEED SPRITES 2068."
Unsolicited Testimonial ... Ron's SON

##
HERE IT IS!

THE TRACER

A NEW PROGRAMMING UTILITY FOR YOUR 2068!

This little package allows you to SEE WHAT YOUR PROGRAM IS DOING! Imagine being able to watch the program lines go by at the bottom of the screen as the display continues unaffected (other than the speed)!
A GREAT LEARNING TOOL!!
You can see what each command does! What better way to learn the basics of BASIC?

YOU SET THE SPEED! AS SLOW OR AS FAST AS YOU LIKE!
SIMPLE COMMANDS - ONLY 3 OF THEM!

We think that TRACER is the best learning and debugging tool we've seen! We think that you will agree! The working part of the program is 100% MC and it is INTERUPT driven!

Another SUPER product from S & K SOFTWARE, the name that brought you KRUNCHER!©

AT ONLY **$14.95**+$1.50ph
THIS ONE IS A MUST FOR ALL PROGRAMMERS!
IT WILL SAVE YOU MORE THAN THAT IN TIME ALONE!

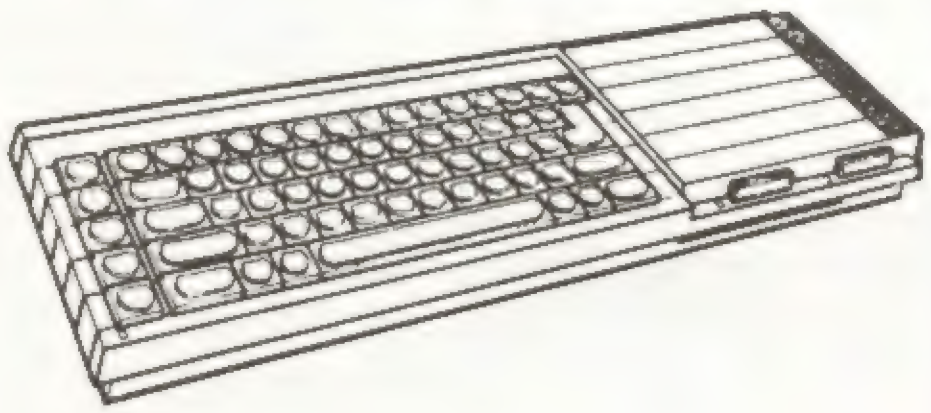
TRACER 2068
IS AVAILABLE EXCLUSIVELY FROM

RMG ENTERPRISES
1419 1/2 7TH STREET, OREGON CITY, OR 97045
503/655-7484
##

C. W. Associates

419 N. Johnson Street

Ada, Ohio 45810



authorized Sinclair QL dealer

SEND FOR OUR FREE CATALOG!
Write or Call:
(419) 634-4874 (6-9pm)

LEMKE SOFTWARE DEVELOPMENT

TS 2068 presents the TS 2068

CHECKBOOK/BUDGET MASTER

The CHECKBOOK/BUDGET MASTER is a checkbook database program and a powerful home or business budget analysis program in one compact/lightning fast program. CBBM was written in BASIC and compiled with the TIMACHINE (by Novelsoft) Basic Compiler!

Program Specification:

Database holds 800 checks (withdrawl or deposits)
Check Recall/Review by: Check Number, Month, Paid To
Functions: REVIEW, LPRINT, DELETE, SUBTOTAL (+ or -)

Budget Analysis: 14 User Defined Categories + 2 (each category has 4 aliases)

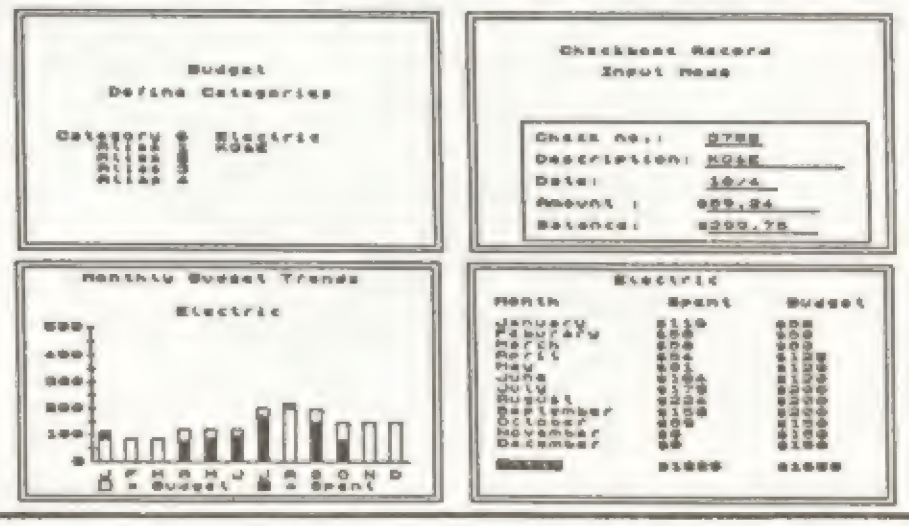
Assign $ Budget to each category (for each month)

Automatically tallies EXPENDITURES as checks are filed and INCOME as deposits are filed!

Update feature: change categories at any time, use the UPDATE command to re-catalog checkbook files

OUTPUT: Tabular or Graphic format
2040 printer or easy full size printer mod.

Get the CHECKBOOK/BUDGET MASTER program from LEMKE SOFTWARE DEVELOPMENT (home of the PIXEL SKETCH and GRAPHICS EDITOR v2.0 the only graphics program to cross the ADVANCED VIDEO MODE barrier providing 32 column, 64 column, and extended color graphics!). CHECKBOOK/BUDGET MASTER and the PIXEL SKETCH and GRAPHICS EDITOR v2.0 are each $19.95 ppd., get yours TODAY!



For a practical demonstration of a chained program, using the Above RAMTOP method of passing data from one module to another, I have chosen one consisting of three modules. A module to set RAMTOP ("RT"), a text entry module ("TE"), and a text LPRINT module ("PRT"). The program will allow you to enter, store above RAMTOP and print out a set of lines consisting of 32 characters per line. The number of lines in a set is available as follows:

USER RAM	LINES PER SET
2K	41
16K	489
32K	1001
48K	1513

With a full 64k memory, the 8k area between the ROM and the system variables area is available and the program could be modified to store an additional 256 lines per set.

In addition to the computer, a tape recorder, and a TS2040 or equivalent 32 column printer, you will need two tapes. Optionally, one of these can be a telephone answering machine message cassette tape--either 60 sec. (RS #43-406) or 3 min. (RS #43-407)--price: $4.95 each. These are both endless loop tapes. If you plan on adding more modules to the program, then purchase the 3 minute tape. I have not found a supplier for a longer endless tape.

Figure No.1 is the program listing for the "RT" module that sets RAMTOP to address 17096. Line 1 is set up to receive a four byte machine code routine to be POKEd by lines 141 to 144, which are then deleted. Line 10 makes the module self running when the program is recorded with the direct command: GOTO 10. Line 20 prevents a TV interupt from occuring during the execution of the program. Lines 30 and 40 POKE the desired address of RAMTOP into the system variable RAMTOP. Lines 50 thru 80 POKE the four addresses under the desired address of RAMTOP with the decimal values that must normally be there (except after a GOSUB and before a RETURN) for the computer to operate properly. Lines 90 and 100 POKE the system variable ERR_SP with the address in line 80 (i.e.; the addresses of the first item on the new Machine Stack). Line 110 calls the machine code routine that is POKEd into the REM statement (Line 1) by lines 141 to 144. Line 120 in conjunction with 140 activates the actual resetting of RAMTOP to the desired address. Line 130 automatically loads the next self running module if the tape is not stopped. Unfortunately, there is no provision in Sinclair BASIC by which you can directly address any of the Z80 internal registers. You must resort to machine code and the USR function. Looking on the left side of page 138 in the TS 1000 or page 142 in the TS 1500 User Manuals, you will find that the Z80 assembler language mnemonic corresponding to the decimal value 49 in line 141 is ld sp,NN. This means load the machine stack pointer registers S and P with the address represented by the values stored in the next two addresses (low byte then high byte). In this case the values 196 and 66 POKEd into addresses 16515 and 16516 by lines 142 and 143. Locating decimal code 201 in the same appendix, you will find that it stands for ret (return). In this case, this returns you to the next line of the BASIC program after the USR function. I hope you noticed that I did not need to use the word hexadecimal until now. Dr. Ian Logan, the leading authority on the Sinclair ZX, TS1000/1500, and Spectrum ROM, states in his book, UNDERSTANDING YOUR ZX81: "The principal behind Hex coding is once again very simple, but it takes a very long time to become fluent in its use, and even programmers of some years experience still have trouble". Because of the interference with the existing GOSUB and machine stacks, RAMTOP must be moved down a minimum of sixty addresses or up a least two addresses using this routine.

Figure No.2 is the listing for the text entry ("TE") module. The program is designed to use as little display as posssible, in order to be able to store more text. Top Down programming was not used in order to locate the text entry loop at the front of the program, adding to the entry speed of text. For the same reason, some memory saving techniques are not used in some places of lines 20 through 80. The dimension for T$ and the value of the variable B, are to be entered by direct commands prior to recording the program. The STOP in line 30 is typed using the shifted A key.

Figure No.3 is the listing for the LPRINT module ("PRT"). The programming is fairly straightforward. The dimension for A$ and the value for the variable B are to be entered by direct commands before recording the module.

As each of the modules will fit in 1k of RAM, in order to save both loading time and tape, set RAMTOP to 17408 before typing in each module. To do this enter:

```
POKE 16388,0
POKE 16389,68
NEW
```

I advise using a regular tape as a master, recording each module with the normal SAVE command before recording it on an operating tape using the GOTO command. Those who elected to use an endless tape as their operating must locate the place where the ends of the tape are spliced with a yellow strip. Never try to rewind an endless tape. The are designed to operate in one direction only. Be sure you turn the sprocket in the correct direction when locating the yellow splice. After locating the splice, make an audio recording, using the built-in mic, of one word only, such as "start" or "one". This will make it easy to locate the splice again should you need to re-record the program. The three modules will just fit on a one minute endless tape without much to spare.

Set RAMTOP to 17408 with the commands given above. Type in the listing of Figure No.1. After checking the program against the listing, record it on the master tape using SAVE "RT". Now use the command GOTO 141 which will poke the machine code into the REM statement. Delete lines 141 to 144. Record the second version of the module on the master tape using the SAVE command. Without rewinding, remove the master tape from the recorder, replacing it with the operating tape and recording the module using the command GOTO 10. When the diagonal LOAD command lines appear on the screen, stop the tape. Without rewinding, remove the operating tape, replacing it with the master.

Again set RAMTOP to 17408. Type in the listing of Figure No.2. Check the program against the listing. Enter the direct commands:

```
DIM T$(32)
LET B= (as listed below)
User RAM                                  B
2K RAM                                    18409
16K RAM                                   32745
32K RAM                                   49129
48K RAM                                   65513
```

Record this module on the master tape using SAVE "TE", then on the operating tape using GOTO 140. When STOP THE TAPE appears on the screen then stop the tape. Replace the operating tape with the master tape.

As the TE module does not change RAMTOP, you can clear the program using NEW. Type in the listing in Figure No. 3 for the "PRT" module. After checking for typing errors, enter the direct commands:

```
DIM A$(1)
LET B=(as listed above).
```

Those using endless tape, replace REWIND TAPE in line 100 with spaces. Save this module on the master tape using SAVE "PRT" then on the operating tape using GOTO 10. When STOP TAPE appears on the screen, then stop the tape.

To operate the program, turn off the computer and then power up. Those not using endless tape must rewind the operating tape. Enter the direct command: LOAD "RT". Then start the tape. When the second module has loaded and STOP TAPE appears on the screen, stop the tape. When the cursor appears on the screen you can start typing in text. The left hand quote symbol marks the end of a 32 character line. After checking the text, use the enter key. Corrections must be made before the enter key is pressed. Spaces to fill out a line need not be typed. Any characters over 32 will be dropped. To stop text entry use the shifted STOP on the A key as the first entry of the next line. Follow the directions on the screen to load the LPRINT module. After the text is printed you have the option of printing another copy or reloading the text entry module to enter a new set of text.

In the CONCLUSION of this series, I will cover how the values for RAMTOP and for the variable B were determined.

```
  1 REM 1234
 10 SAVE "RT"
 20 FAST
 30 POKE VAL "16388",VAL "200"
 40 POKE VAL "16389",VAL "66"
 50 POKE VAL "17095",VAL "62"
 60 POKE VAL "17094",VAL "0"
 70 POKE VAL "17093",VAL "6"
 80 POKE VAL "17092",VAL "118"
 90 POKE VAL "16386",VAL "196"
100 POKE VAL "16387",VAL "66"
110 RAND USR VAL "16514"
120 GOSUB VAL "140"
130 LOAD "TE"
140 RETURN
141 POKE 16514,49
142 POKE 16515,196
143 POKE 16516,66
144 POKE 16517,201
```

FIGURE NO. 1

```
 10 INPUT T$
 20 FOR N=1 TO 32
 30 IF T$(1)=" STOP " THEN GOTO
VAL "90"
 40 POKE A,CODE T$(N)
 50 LET A=A+1
 60 IF A=B THEN GOTO VAL "90"
 70 NEXT N
 80 GOTO 10
 90 POKE A,VAL "227"
100 PRINT AT SIN PI,SIN PI;"PRE
SS ANY KEY THEN START TAPE"
110 PAUSE VAL "32768"
120 CLS
130 LOAD "PRT"
140 SAVE "TE"
150 PRINT AT SIN PI,SIN PI;"STO
P TAPE"
160 PAUSE VAL "120"
170 PRINT AT SIN PI,SIN PI;"ENT
ER TEXT"
180 LET A=VAL "17097"
190 GOTO VAL "10"
```

FIGURE NO. 2

```
 10 SAVE "PRT"
 20 PRINT AT SIN PI,SIN PI;"STO
P TAPE"
 30 PAUSE VAL "120"
 40 PRINT AT SIN PI,SIN PI;"PRI
NT TEXT? ENTER Y/N"
 50 INPUT A$
 60 CLS
 70 IF A$="N" THEN GOTO VAL "10
0"
 80 IF A$="Y" THEN GOTO VAL "14
0"
 90 GOTO VAL "40"
100 PRINT "REWIND TAPE,PRESS KE
Y,START TAPE"
110 PAUSE VAL "32768"
120 CLS
130 LOAD "TE"
140 FOR N=VAL "17097" TO B
150 IF PEEK N=VAL "227" THEN GO
TO VAL "180"
160 LPRINT CHR$ PEEK N;
170 NEXT N
180 LPRINT
190 GOTO VAL "40"
```

FIGURE NO. 3

Understanding And Upgrading The TS1016 RAM Pack

by Tim Stoddard

This is the second part on upgrading your TS 1016 RAM Pack to 64k. Last issue we discussed the ins and outs of dynamic memory and how the Sinclair RAM Pack works. This issue it's time to warm up the soldering irons!

Take a look at Fig.1. You'll note that the circuit schematic looks quite similar to the one in the last issue. There are, however, some significant differances. The biggest change is the addition of selection logic (the 74LS138, 74LS139). Missing is the noisy DC to DC converter that generated the +12 and -5 volt bias voltages needed by the older 16k DRAMS.

Another more subtle change is the addition of the active low OR gate in address line 15. This brings up the unusual architecture used in the ZX/TS machine. The interupt routines in the Sinclair ROM ASSUME the display to be under the 32k boundry! So if y;ou add enough memory to extend beyond the 32k boundry and then initialize it, you will lose the display! To get around this problem we must force the memory to "look" like 32k during an interupt cycle. This is done by oring A15, the address bit that determines which 32k boundry were in, and M1 which occurs during an interupt cycle. Unfortunately the M1 cycle also occurs during EVERY instruction fetch. The effect of this is that you CAN NOT EXECUTE PROGRAMS ABOVE 32K. However, you CAN store data, such as a large array above the 32k boundry which is what most people want the extra memory for anyway...So, warm up the old soldering iron an let's go...

The conversion is done in two steps and should take someone with "good" experience a weekend to complete. I should point out at this time that neither myself not Time Designs Magazine is responsible for any damages caused to your RAM Pack or your computer by this modification. THIS IS NOT A GOOD FIRST OR EVEN A TENTH PROJECT. You'll need experience in PCB repair and handling a low power soldering iron. I will assist anyone having trouble by either BBS communication (CompuServe ID 73127,2664; Zebra BBS ID "Tim"), or S.A.S.E. mail from you (85-48 66th Road, Rego Park, NY 11374). I would recommend, if your not too confident, that you purchase a 16k RAM Pack from Zebra Systems or other source, to modify. They are inexpensive (under $10) and will allow you to use your ZX/TS while taking a break from the modifications.

A WORD ABOUT STATIC ELECTRICITY: Very simply, it can destroy all the work you put into a project in just a few nano-seconds. Work on an anti-static mat. This can be a commercial item or a piece of aluminum foil. The idea is to keep you, the project, and anything that touches the project at the SAME POTENTIAL. Use an ungrounded tip type soldering iron.

You'll need the following PARTS:

(8) 4164 or equivalent 64K DRAMs
(1) 74HCT138 or 74LS138
(1) 74HCT139 or 74LS139
(1) 74HCT00 or 74LS00
(8) 16 pin IC sockets
(1) 1K 1/4 watt resistor
(11) 1N4148 or 1N914 diodes

You'll need the following TOOLS:

23 watt soldering iron
solder sucker/wick
small wire cutters (Xcelite 73CG is ideal)
small needle nose pliers (Xcelite 79CG is ideal)
30 gauge wire-wrap wire
20-24 gauge solid wire
Dremel moto-tool with extra-small ball cutter or an Xacto knife
Crazy glue
solder
Anti-static mat

F I V E V O L T D R A M C O N V E R S I O N

1) Dissasemble the case on your anti-static mat. From this point on BE CAREFULL with the ribbon cable connecting the two PCBs, it is very easy to break a wire in it and not even know it 'till you have powered up.

2) Remove all componants from the DRAM PCB not marked in Illustration "A". Start with the small componants first by using the solder sucker/wick to remove the solder from the pad and then using the needle-nose pliers to work the wire loose. TAKE YOUR TIME! When you get to the DRAM ICs use this method: take the small wire cutters cut all the leads on one side of the IC close to the PCB, then bend the IC up then back & forth to break off the leads on the other side of the IC. Now use your solder sucker/wick to remove the solder and old IC lead from each of the pads. WORK VERY CAREFULLY HERE. DON'T LIFT ANY OF THE FOIL PATTERNS. Take a break after each DRAM removed.....you'll be rewarded with good clean job, and a ram pack that works!

4) Check the DRAM PCB for solder splashes, shorts, etc. At this point you should only have 6 de-coupling caps and 1 electrolytic cap left on the board.

5) Install the eight 16 pin sockets in the DRAM locations placing pin 1 toward the electrolytic cap.

6) Install jumper "A" where a cap used to be as shown in Illustration "A". This jumpers one of the multiplexed address lines to ground to make the ram pack a 16K version. This jumper will be removed later, after testing.

7) Make the 3 cuts, and 3 adds as shown in Illustration "B".

8) Carefully install the PCBs onto the computer (leaving them out of the case), and power up. If all is well you should get the usual "K" cursor in just a few seconds. Check to see if the ram was properly initalized by executing the following command: PRINT PEEK 16388 + 256 * PEEK 16389. You should get 32768. If not re-check the above steps and find where you went wrong?

This completes the 5 volt conversion step.

S I X T Y F O U R K C O N V E R S I O N

1) Your ram pack should be fully operational as a 16K pack using the 5 volt only 64K DRAMS at this point. DO NOT CONTINUE ON UNTIL THIS IS TRUE.

2) Perform the cuts and adds as shown in Illustrations "C" & "D".

3) Take the three ICs (74LS138,139,00) and bend all leads horizontal from the body except the power leads (pins 8,16 for the 74LS138,139 and 7,14 for the 74LS00). See Illustration "E".

4) Using Crazy glue, and working VERY FAST glue the 74LS138, lining up the power leads on top of IC "A" the 74LS157 on the CONTROL PCB (the PCB with the connector on it). See Illustration "G" for IC identification. Next glue the 74LS139 lining up the power leads again to the 74LS138 just glued on. Finally glue the 74LS00, lining up it's pin 14 to the 74LS139's pin 16.

5) Carefully bend back pin 7 on the 74LS00 (top of IC stack) so that it touches pin 8 of the 74LS139 under it. After insuring all the power leads are lined-up and touching, solder them. Check with Illustration "E".

6) Using Illustration "F" and 30 gauge wire-wrap wire:

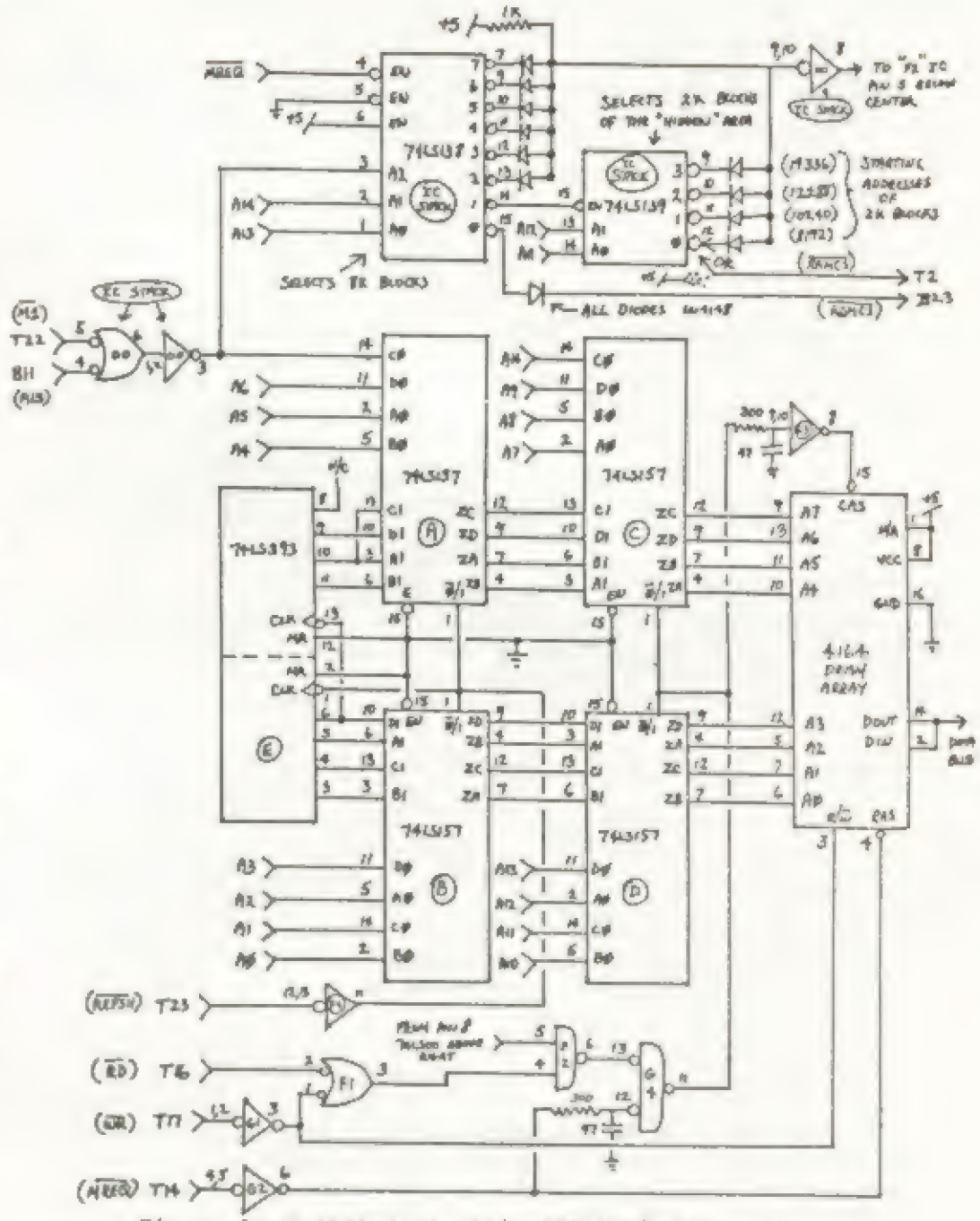


Figure 1: Modified Sinclair RAM Pack Schematic

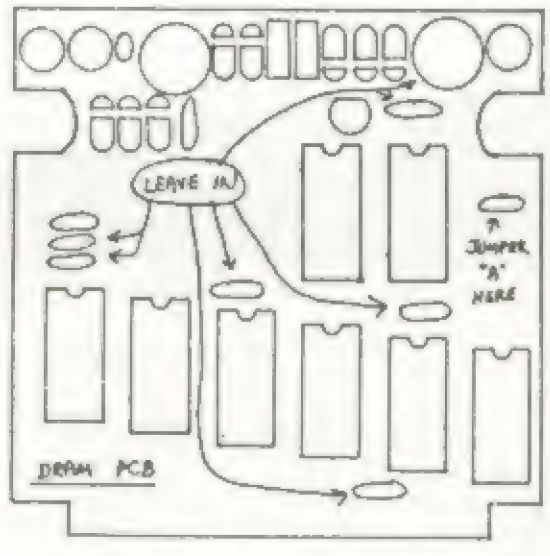


Illustration A: Component Side

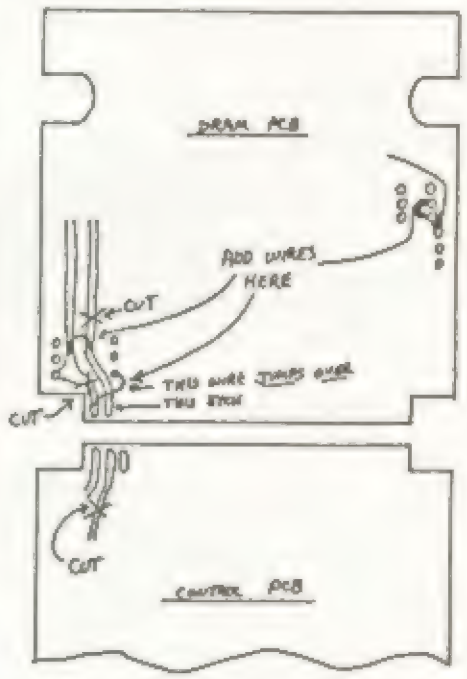


Illustration B: Back Side

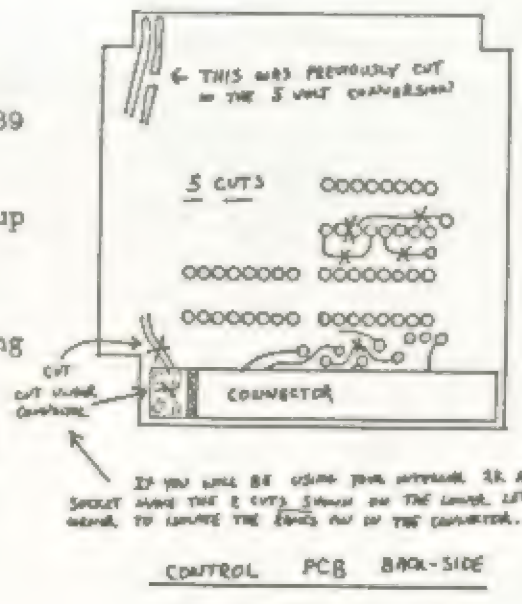


Illustration C: Cuts

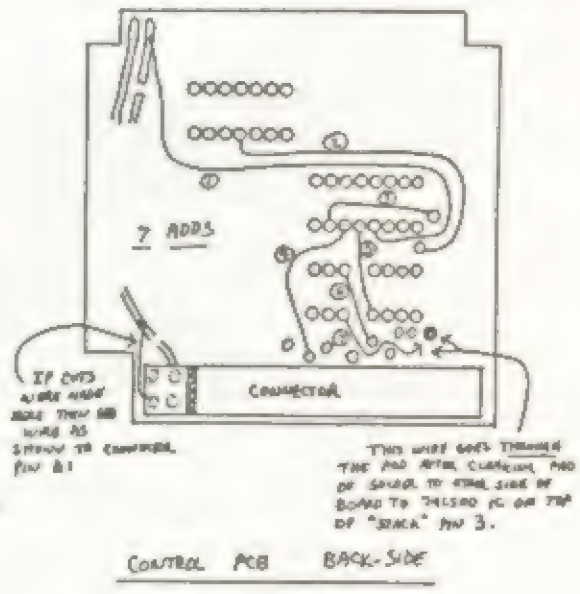


Illustration D: Adds

ADD WIRE FROM HERE	TO HERE
74LS00, PIN 3	74LS138, PIN 3
74LS138, PIN 14	74LS139, PIN 15
MREQ on connector	74LS138, PIN 4
74LS138, PIN 8	74LS138, PIN 5
74LS138, PIN 16	74LS138, PIN 6
A14 on connector	74LS138, PIN 2
A13 on connector	74LS138, PIN 1
M1 on connector	74LS00, PIN 5
A15 on connector	74LS00, PIN 4
74LS00, PIN 6	74LS00, PINS 1&2
A12 on connector	74LS139, PIN 13
A11 on connector	74LS139, PIN 14

7) Add a prepared diode with the anode soldered to pin 15 of the 74LS138. Then add a 30 gauge wire from ROMCS on the connector to the cathode of this diode.

8) Add five prepared diodes with the cathodes soldered to pins 9,10,11,12,& 13 of the 74LS138. Then add a NON-PREPARED diode with the cathode soldered to pin 7 of the 74LS138. Bring the diode around the IC "stack" and line up it's anode with the other 5 diodes. solder all six anodes forming a "buss". See Illustration "F".

9) Next solder a 1K resistor from pin 16 of the 74LS138 (+5 volts) to the "anode buss".

10) Add diodes in the following table for each of the 2K blocks of 8K "hidden" area that you want to use.

RAM AREA RANGE	CATHODE TO PIN OF 74LS139
8192 to 10239	12
10240 to 12287	11
12288 to 14335	10
14356 to 16383	9

Tie the anodes of any of the diodes used above to the "anode buss".

Add wire from the "anode buss" to pins 9 & 10 of the 74LS00. Then add a wire from pin 8 of the 74LS00 to the pad shown in Illustration "G" (this pad runs to pin 5 of the 74LS00 IC "F" on the CONTROL PCB.

Remove jumper "A" in Illustration "A".

11) Plug the ram pack onto the computer and power up. If all is well you should get your "K" cursor. Execute: PRINT PEEK 16388 + 256 * PEEK 16389. This should give you 32768. If this works enter the following command lines one at a time:(1) POKE 16388,255 (2) POKE 16389,255 (3) NEW (4) PRINT PEEK 16388 +256 * PEEK 16389. You should now get 65535! indicating that the entire ram is now initalized and ready for use.

12) re-assemble the PCBs back into the case and re-test as above. This completes the conversion.

OPTIONS: You can use the internal RAM socket via the RAM Pack selection logic. This is where I placed my ZX-LR8 ROM for high speed cassette access. The cuts for this option are shown in illustration "C", and the adds are shown in Illustration "D". Those cuts and adds just isolate the RAMCS pin on the connectior from the +5 volt buss it was normally connected to (the RAM Pack normally disables the internal 2k RAM). Illustration "F" then shows where to connect the wire to use the RAMCS pin to enable the internal RAM socket. Note that you could use any of the 2k selection blocks from the 74LS139 chip. See the schematic (Fig.1).

Another great option is the ability to change the configuration of the RAM Pack via a DIP switch. On one of my prototypes, I installed a DIP switch to allow enabling or disabling any of the four 2k blocks in the 8k "hidden" area. The best physical location is shown in Illustration "F". The way I electrically connected it is shown in the schematic of Fig.2. You could also use the switch arrangement to enable or disable any ot the 8k system blocks too. In fact, Fig.2 shows a combination of switching both the 2k "hidden" blocks and the 8k system blocks. After you glue the switch in place, you can cut a small access hole in the side of the case with the X-acto knife so you can change the configuration without taking apart the case.

That's about it. Write and let me know how you made out. I've also designed from the ground up an expansion RAM that uses the new 256k RAMS (64k by 4 bit). The entire circuit uses just 9 chips and takes advantage of the newer DRAM's internal refresh logic. If there is enough interest, I'll submit the article to TDM.

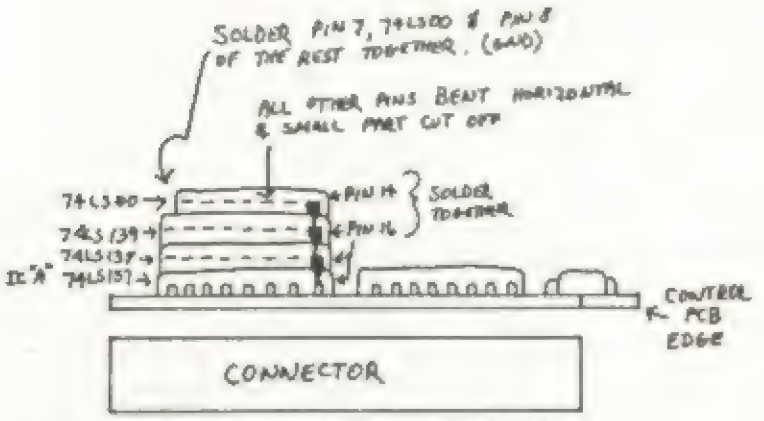


Illustration E: IC Stack

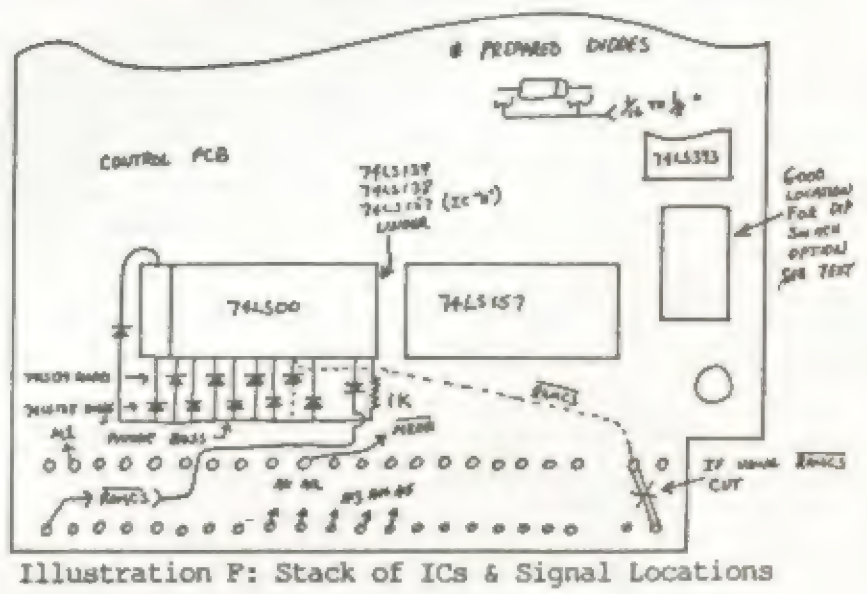


Illustration F: Stack of ICs & Signal Locations

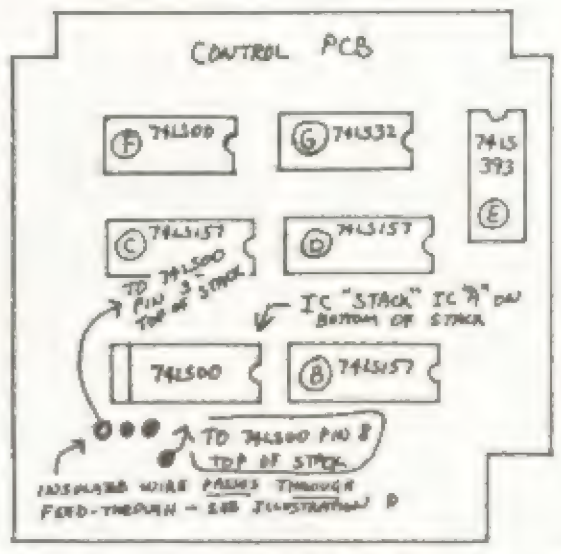


Illustration G: Pad/Feed-Through Locations

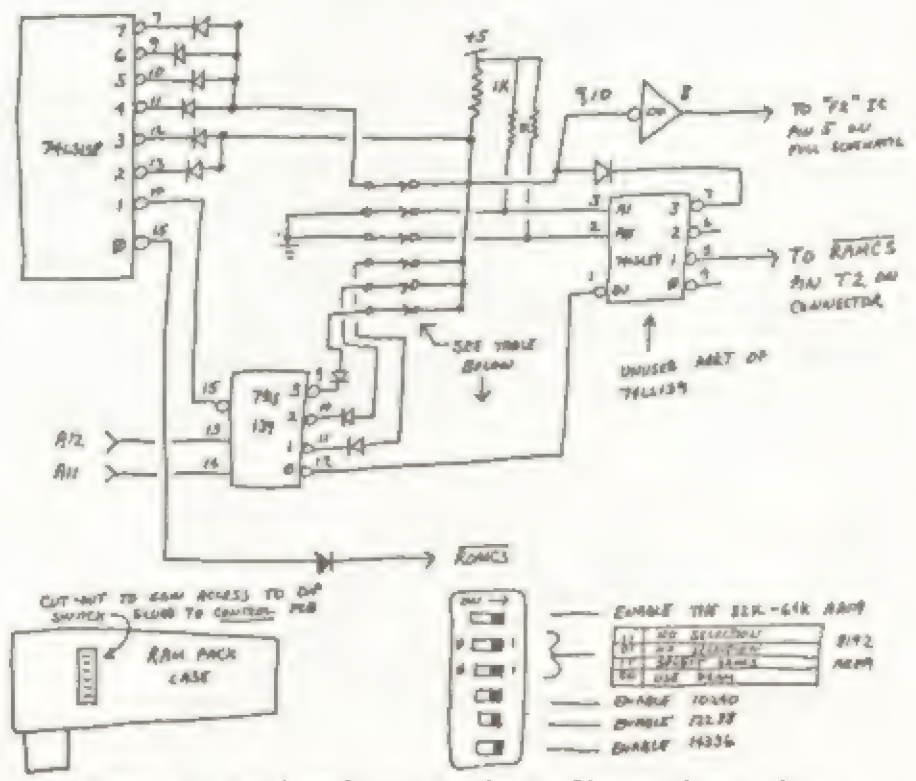
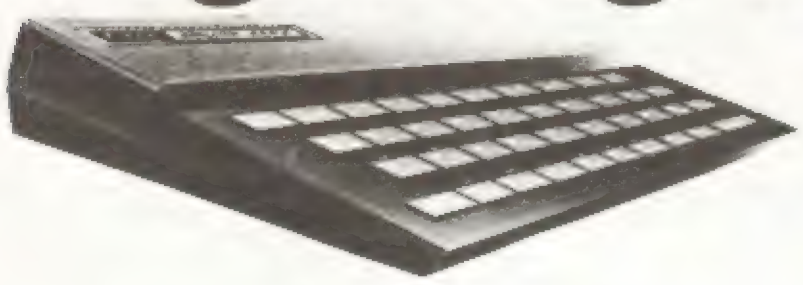


Figure 2: Optional RAM Pack Configuration Switch

Beginning Z80 Machine Code



LESSON FIVE

By Syd Wyncoop

I left the last lesson with a challenge to you to rewrite the sample disassembly from Lesson 2 to eliminate the overflow error it contained. If you had difficulty, refer to Lesson 4. The answer was given in the comparison which explained the ADC instruction. How many of you thought of rewriting the routine using the sixteen bit intructions? Did you use LD HL,(pq) and LD BC, (pq)? Can you see how a short Basic interface (program) could collect the values and call the MC routine to perform the addition? I trust some of you are beginning to have some ideas.

We know how to load a register (pair) of memory location and perform arithmetic with the values loaded. We would, however, find MC of very limited value if these were all it could do. Most of you are familiar with the Basic commands GOTO and GOSUB. In truth, it is these instructions that give a program the power to do some real work for us.

In MC, the equivalent instructions are referred to as Jumps and Calls. The syntax for these instructions are given in Chart 5. You will note a new abbreviation, c, which is a test for the condition (or status) of a flag.

We briefly discussed the Carry flag last lesson. Here is how the F (flag) register is arranged:

```
Bit#   7  6  5  4  3  2  1  0
Flag   S  Z  .  H  .  P/V  N  C
```

Where:
- S = Sign
- Z = Zero
- H = Half-Carry
- P/V= Parity/Overflow
- N = Subtract
- C = Carry
- . = Not used

Sign Flag - Stores the sign of the last result. Flag will be set for a negative result and reset for a positive result (always reflects the most signifigant bit of the result).

Zero Flag - Checks whether last result was zero. Flag will be set if result is zero, else reset. Note: flag = 1 if result = 0. Watch it!

Half-Carry- Used internally by CPU to record carry from bit 3 to bit 4 in registers or bit 11 to bit 12 in register pairs. We will ignore it.

Parity/Overflow- Has two jobs depending on the instruction last executed.
Parity is the number of set bits in the result and is referred to as odd or even. Flag will be set if parity is even and reset if odd. Note: even parity generates an odd flag. Watch this one, also!
Overflow records a carry from bit 6 into bit 7 which effectively changes the sign of result in signed arithmetic operations. Flag will be set for overflow, else reset.

Subtract Flag- Used internally by CPU to record whether last instruction was addition or subtraction. Flag will be set if was subtraction operation. We will ignore this one, also.

Carry Flag- Our old friend records a carry from bit 7 to bit 8 in registers or bit 15 to bit 16 in register pairs. Is also used to save the lost bit in the shift and rotate instructions.

You will note that two bits of the flag register are unused. The status of these bits are important and there are no instructions that affect them.

Each flag can be in one of two states...set or reset (on or off). A set bit = 1 (on) and a reset bit = 0 (off). This can become very confusing when using the Zero or Parity/Overflow flags, as the flag will not be as we expect it. For instance, the Zero flag = 0 if the result was not zero. Most of the time, however, you can use the flags without knowing whether they are set or not. You need only test their status and jump accordingly.

Each flag indicates a specific condition based on the result of the last instruction executed. Chart 6 indicates how the flags are affected by the various instructions. It is important to know how the flags are affected as every instruction does not affect them and many instructions do not affect them as you might expect.

Enough of that, back to the Jump instructions. This instruction has two versions, Jump and Jump Relative. The mnemonics are JP and JR, respectively.

JP is equivalent to Basic's GOTO. JP begins executing the next instruction at the absolute address you specify as its argument. A JP 4000h instruction will send the CPU off to address 4000h to find the next instruction to execute. Your umps can be conditional...that is, they can test one of the flags and jump only if the condition is met.

JR requires the introduction of another Hex to Decimal conversion chart, Chart 7. You will note that the first half of this chart is the same as our previous Hex to Dec chart (Lesson 1). The last half, however, indicates negative numbers. When numbers are used in this fashion, they are referred to "signed numbers". Signed numbers merely means that the most significant bit (bit 7) is used to represent the sign of the number. A set bit (1) is a negative number and a reset bit (0) is positive.

JR also requires a brief discussion of the register pair PC. PC is a special register pair not normally accessible to us. It is called the Program Counter and its job is to keep track of where the next instruction to execute is located. All Z80 instructions are 1,2,3 or 4 bytes in length. The CPU will always advance PC by the correct number of bytes for the instruction it is about to execute. The effect of this is to skip any arguments belonging to the current instruction so as to be in position to fetch the next instruction.

Any jump instruction causes PC to discard the address it contains and replace it with the new address, as specified in the jump instruction. Note, PC will always contain the address of the next instruction to execute, not the current one.

The JR instruction adjusts the PC by adding the value specified to the current value of PC. In other words, JR tells the CPU to Jump to address X, which is Y bytes from where PC is. Y can only be in the range of -128 to 127 and X is the calculated new address. In the case of negative values, the program would jump back to a previous instruction (loops) while positive numbers would cause the skipping over of the next Y bytes.

JR can also be conditional as indicated in Chart 5 and discussed above for JP.

When programming in Basic, it is quite common to have a line such as:

```
100 GOTO 10*VAL A$+1000
```

There is a MC instruction, JP (HL), which emulates this type of operation. This instruction will jump to the address held in the HL register pair. This allows a routine to build up an address from tables or inputs and transfer program control to that address. We will not discuss this much further now as it represents some pretty advanced programming.

CALL is our GOSUB equivalent. It acts exactly like BASIC's GOSUB. A jump is made to the specified address and a return is made to the instruction that would have been executed next had the CALL not been encountered. This is accomplished by saving the address in PC on the stack (we will explain the stack later) before making the jump.

There is a special case of CALL, that does not require an address to be specified, which is know as RST. RST is read restart , and is unique because it is the only instruction that uses an eight bit address. RST calls a subroutine with a one byte instruction.

Some important points about RST are that it is unconditional and usually computer specific (can not run on another 280 based computer). Being computer specific is due, unfortunately, to there already being instructions at all the RST addressed, which cannot be changed. This is due to our operating system being in a ROM type memory. All is not lost though. Since these are very handy instructions, Sinclair put some of the most accessed routines there. We will find that we can use some of the RST instructions, after all.

As with any GOSUB instruction, Calls and RSTs require a return instruction to let the CPU know the routine has finished its task. The mnemonic for return is amazingly enough RET. RET will perform exactly the operation you would expect it to, and

your returns can be conditional. Conditional returns allow for many exit points based on completing certain tasks. There are two special RETs which we will discuss later because they are used to return from the interrupts.

We have learned about the flags and how to make jumps and calls based on their status. We now need to explore some of the ways to set these flags in order for our tests to be meaningful. One of the ways to do this is directly with the CCF and SCF instructions.

CCF means Complement the Carry Flag. If Carry was set, it will be reset and vice versa. SCF means Set the Carry Flag. The Carry flag will set by this instruction.

Another way to affect the flags is with the remainder of the arithmetic instructions (I've been holding out on you again). These are also listed on Chart 5, and can not truely be refferred to as arithmetic instructions, except for CP.

CP, which means Compare, is a neat and often used instruction. CP sets all the flags as if a value were subtracted from the Accumulator, but without changing the value of the Accumulator! It is important to realize the result of the Compare is not stored anywhere, only the flags are affected. CP has two special forms, CPI and CPD, which are read Compare with Increment, and Compare with Decrement. CPI performs the same as a CP (HL) instruction would, except that HL is incremented and BC is decremented. The only flag affected is the P/V flag which is set according to the value of BV. If BC = 0, then P/V = 0.

CPD is the same as CPI except that HL is decremented. The effect on the flags is the same.

The next instruction is DJNZ...which is not Greek! DJNZ is read "decrement the B register and jump relative if B is not zero". This is an extremely useful instruction which leads to the B register being used as a counter. DJNZ can be compared to the Basic loop control variable. The equivalent Basic statement would be as follows:

```
10 For X = 10 to 0 Step -1
20 (do job here)
30 Next X
```

In order to perform the same operation as DJNZ using any other register, you would need two instructions:

```
DEC L
JR NZ, Loop
```

To use DJNZ, you must properly load the B register. You can then construct a loop to do whatever task you wish. You can even reuse the B register in the loop, if you properly preserve its value first. More on this preservation of values later.

CPL stands for Complement. Each bit of the Accumulator is altered (complemented). For example: if the Accumulator contains 11011101b, its complemented form would be 00100010b.

NEG is the last unexplained instruction on chart 5. NEG will negate the Accumulator, which means to place the two's complement of the A register in the Accumulator. if the Accumulator contains 5, it will be negated to -5.

You now have about one third of the Z80 instruction set, and with the stack instructions next issue (they are certainly the most used of the instructions). You are now armed with the tools to write a MC program of your own design. I encourage you to experiment and see if you get the desired results. I will reply personally to all enquiries that contain a S.A.S.E., if you have difficulty (send to- 2107 S.E. 155th St., Portland, OR 97233).

With the next lesson, we will explore printing to the screen as that will give us some immediate feedback as to how we are doing and whether our routine is working. If you have any information on the display file and/or ROM routines, you should review it, in anxious anticipation.

CHART 5

Jumps	Flag setting
JP nn	CCF
JP c,nn	SCF
JP (HL)	
	CP n
JR e	CP r
JR c,e	CP (HL)
DJNZ e	CPI
	CPD
CALL nn	
CALL c,nn	CPL
RST xx	
	NEG
RET	
RET c	

Where: n = any numeric constant 0 to 255
nn = any numeric constant 0 to 65535
r = any single register
e = any numeric constant -128 to 127
c = flag status
xx = 00h, 08h, 10h, 18h, 20h, 28h, 30h, or 38h

CHART 6

Instruction(s)	C	Z	P/V	S	N	H	Comments
ADD, ADC	*	*	V	*	0	*	8 bit add or add w/carry
ADD	*	-	-	-	0	-	16 bit add
ADC	*	*	V	*	0	-	16 bit add w/carry
AND	0	*	P	*	0	1	Logical operations
BIT	-	*	-	-	0	1	Specified bit copied into zero the flag
RES, & SET	-	-	-	-	-	-	Bit instructions
CCF	*	-	-	-	0	-	Complement carry flag
SCF	*	-	-	-	0	0	Set carry flag
CP, NEG, SUB, SBC, DEC, & INC	*	*	V	*	1	*	8 bit subtract or subtract w/carry, compare or negate accumulator & 8 bit decrement
DEC, & INC	-	-	-	-	-	-	16 bit decrement and increment
SBC	*	*	V	*	1	-	16 bit subtract w/carry
CPI, CPIR, CPD, & CPDR	-	*	P	-	1	-	Block searches; Z=1 if A=(HL), else Z=0;P/V=1 if BC not equal to 0, else P/V=0
CPL	-	-	-	-	1	1	Complement accumulator
DAA	*	*	P	*	-	*	Decimal adjust accum.
IN	-	-	-	-	-	-	Input register direct
IN	-	*	P	*	0	0	Input register indirect
INI, IND, OUTI, OUTD, INIR, INDR	-	*	-	-	1	-	Block in & out instructions; Z=0 if B is not equal to 0, else Z=1
OTIR, & OTDR							
LD	-	-	-	-	-	-	Assignment instructions
LDI, LDD, LDIR, & LDDR	-	-	P	-	0	0	Block transfers; P/V=1 if BC is not equal to 0 else P/V=0
OR, & XOR	0	*	P	*	0	0	Logical OR accumulator
RLA, RLCA, RRA, & RRCA	*	-	-	-	0	0	Rotate accumulator
RL, RLC, RR, RRC, SLA, SRA, SRL	*	*	P	*	0	0	Rotate and shift left or right

Where: * = Flag changed according to result
- = Flag either unchanged or undeterminable
0 = Flag reset
1 = Flag set
P = Parity changed according to result
V = Overflow changed according to result

CHART 7

Signed Numbers--Hex/Dec Conversions

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	,8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	-128	-127	-126	-125	-124	-123	-122	-121	-120	-119	-118	-117	-116	-115	-114	-113
9	-112	-111	-110	-109	-108	-107	-106	-105	-104	-103	-102	-101	-100	-99	-98	-97
A	-96	-95	-94	-93	-92	-91	-90	-89	-88	-87	-86	-85	-84	-83	-82	-81
B	-80	-79	-78	-77	-76	-75	-74	-73	-72	-71	-70	-69	-68	-67	-66	-65
C	-64	-63	-62	-61	-60	-59	-58	-57	-56	-55	-54	-53	-52	-51	-50	-49
D	-48	-47	-46	-45	-44	-43	-42	-41	-40	-39	-38	-37	-36	-35	-34	-33
E	-32	-31	-30	-29	-28	-27	-26	-25	-24	-23	-22	-21	-20	-19	-18	-17
F	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

SALE!★

640K QL . . . $ 399!+

QL, with 4 Psion programs, 4 blank mdv cartridges, and 640K RAM $399.00+
512K RAM PAK for QL, brings total memory to 640K..... $199 PP!
AVATEX 300/1200 BAUD MODEMS, 8 LED, RS232, etc..... $109 PP!
AVATEX 300/1200, 100% Hayes Compatible..... $159 PP!
QL MODAPTER, allows use of 300 Baud modem with QL, includes QCODE terminal program which supports up/downloading, etc..... $ 59.95+
Modapter WITH AVATEX 1200 HC (Hayes Compatible)..... $209 PP!
QL ASSEMBLER WORKBENCH, includes monitor, disassembler, assembler, by Talent, a great buy, great software! Only..... $ 49.95+
TRANSFORM CASE, plastic case holds 20 MDV cartridges..... $ 5.99+
MICRODRIVE CARTRIDGES for QL, ZX Microdrives ONLY(4 carts).. $ 10.95+
QL BOOKS, Assorted, call for titles, only..... $ 9.95+

THE 2068 is NOT forgotten !!

$$$ SALE $$$>>> Machine Code Tutor, Fighter Pilot, Night Gunner, or Loader V, only $10 each POST PAID while current supply lasts!

SPRITES 2068, all new! Design, combine, and use sprites quickly and easily. Includes GREAT demos! Includes detailed manual, sprite service utility (2520 bytes MC), spritedraw program..... $ 19.95+

SABOTEUR, you are a ninja spy who must get a disk from the enemy wharehouse; use lethal kicks, punches, stars, knives, grenades to 'remove' guards and dogs who 'get in the way'. Super graphics! $ 15.95+

CRITICAL MASS, aliens threaten to destroy your solar system by taking over your anti-matter conversion plant. You must infiltrate the plant and destroy the converter before BOOM! and it turns into a black hole, destroying all stars and planets in your system..... $ 15.95+

SPECTERM 64- Brand new 64 column terminal software for the Spectrum or 2068 with Spectrum Emulator and 2050 modem OR ANY RS232 BOARD! Yes, 1200 BAUD (300 baud with 2050 modem) on a 2068! You get 64 columns ON SCREEN, built in XModem up and downloading, half/full duplex, 31K buffer, can transmit/receive ANY file- Bytes or Program. Automatically saves Tasword II files properly! 4K user menu area, and documentation for full power modeming, all for only..... $ 29.95+

DOUBLE SIDE/DOUBLE DENSITY floppy disks, highest quality, lifetime guarantee, Tyvec sleeves, user ID labels, and write protect tabs, only 75 CENTS each (minimum 10), or 100 for 65 cents each! (100 shipped PP!)

75¢ DISKS

Unless otherwise noted, please add $3 shipping for software, $5 for hardware, or $10 for a QL package.

Send check or Money Order to:

VARIETY SALES COMPUTER DIVISION
Department TDN-3
325 WEST JERSEY STREET, #2D
ELIZABETH, NEW JERSEY 07202
201-527-0535 FOR MORE INFORMATION.

SOFTSYNC TS2068 Inventory Liquidation Sale

Zebra Systems just made a special purchase of all the Timex/Sinclair product left in Softsync's warehouse in order to bring you these fine programs at special low prices. Softsync started in the Timex market with excellent products, but they now focus on IBM, Commodore and other large markets, where they sell products like the Personal Accountant for much more money.

All product is new and packaged with documentation in Softsync's attractive four-color boxes. Quantities are limited, so act now. These products list at $19.95 to $24.95 and are in our regular catalog for $17.95 to $22.95. But now our liquidation sale prices are as follows:

INVENTORY SALE PRICES:

1 for $10	1st $10
2 for $19	2nd $9
3 for $27	3rd $8
4 for $34	4th $7
5 for $40	5th $6
6 for $45	6th $5

Zeus Assembler Cat# SS05

Machine Code programming is made simple with Zeus. This sophisticated programmers' aid allows the use of the full Z- 80 mneumonic instruction set and comes with a step by step instruction booklet.

Unique features include: a full screen editor, automatic line numbering and renumbering and mini monitor. Displays current registers and single stroke commands save machine code.

Zeus Monitor and Dissassembler Cat# SS06

Acclaimed by reviewers as the most comprehensive Monitor and Dissassembler available.

This powerful programming tool allows you to translate machine code into comprehensible assembly language instructions, enabling you to examine the BASIC ROM, to investigate the workings of the Timex/Sinclair 2068 or to analyse your own machine code routines.

With the highly versatile Monitor, you get an extensive set of facilities to aid the entry, inspection, modification and debugging of your own machine code programs.

CYBERZONE Cat# SS02

Special Feature: Use your voice to activate your laser fire! The sound of your voice sends lasers shooting at the enemy.

Imagine yourself pitted against the Cyber's ultra accurate laser fire. The situation is tense as you avoid the swooping, spinning fighters coming to get you.

Can you stay alive long enough to chip away at the floor of Cyber's spinning spacecraft? And is your aim deadly enough to hit the Cyber's only vulnerable spot...his left foot?

Cyberzone is a fast game with five levels of play and exciting graphics that place you in the center of the action.

Personal Accountant #SS04

The Personal Accountant is a powerful yet purposefully simple accounting program for household and small business use. Using a time honored accounting practice known as "Double Posting Book-keeping" the Personal Accountant will instantly organize all your financial information.

Open as many accounts as you need, balance your checkbook instantly, track loans, charge card purchases, expenses and IRA's. Generate financial reports ranging from trial balances to profit/loss, expense vs. income and assetts over liabilities. An amortization table can calculate payments and changing rates, generate future value and growth tables. And a built in data base keeps names, addresses and other vital data at your fingertips.

The Personal Accountant is comprehensive yet simple to use with no codes to memorize and the screen will guide you every step of the way.

VOICE CHESS Cat# SS01

The Most advanced chess game available for the Timex/Sinclair 2068 actually talks to you during the game. A digitized voice speaks through the computer's speaker, advising of its move, recommending moves for you and making facetious comments.

Voice Chess is written in fast machine code so it responds to your moves quickly.

Features include: analyze mode, recommend move, change colors or levels at any point in the game, save, reload and print out any game you play. Displays full Chess board in detail.

GULPMAN Cat# SS03

The cursed wormoids are out to get control of Gulpland, chasing its inhabitants out of their apple orchards. Eat as many apples as you can to get bonus points and use your lasers to stun the wormoids. 15 different mazes.

Ordering Instructions: Include $3.00 S&H. VISA/MC Accepted.

Zebra Systems, Inc.

78-06 Jamaica Avenue
Woodhaven, NY 11421
(718) 296-2385

Dear Time Designs Reader,

We are very pleased to bring you our Zebra Catalog as the centerfold of the November/December Time Designs Magazine. For economic reasons the catalogs were printed previously in one large batch and have become slighted dated. For that reason we are supplementing it with several pages of new product announcements, sales, and a short list of updates.

Particularly exciting are the new bargain prices we are able to offer on Softsync's very high quality software products, and on our TS2068 compatible Trackball controllers. We will continue to strive to bring you the best support products for Timex computers and at the best prices.

Please accept our best wishes for the Holiday season and for a happy and healthy New Year.

Sincerely yours,
Jane, Linda, Jeff, Tom, and Stewart
The Staff of Zebra Systems

Zebra Best Sellers

For those who are interested, here is a short list of our currently best selling products to check out in our Time Designs ads and on the various pages in our 1986B Catalog.

TS2068 Trackballs (TD)
Sprites 2068 (TD)
Greeting Card/Banner/Sign Designers (P3)
TS2068 Technical Manual (P1)
Tech Draw Jr.(P2)
OS64 Cartridge (P7)
Mscript (P4)
64K TS1000 RAM PACKS (P13)
Machine Code Test Tool (P15)
Discounted Books (P10-11)
TS2050 Modem Boards (P12)

TS2068 Trackball Only $19.95

Originally sold for $69.95
Specify Cat# TBTMX02

Plugs into TS2068 Joystick Port and works with all joystick software.

Bonus Feature: Also works on Commodore 64, VIC-20, ATARI 800, and more. Contact factory for more complete list.

You can benefit from our recent purchase of brand new WICO Trackball Controllers at closeout prices. We've taken the model WICO originally made for the Texas Instrument 99/4A and made a very simple modification so it now is fully compatible with the Timex TS2068's joystick port.

WICO is the largest designer and manufacturer of control devices for commercial arcade video games. If you've ever played an arcade video game, chances are you've used a WICO joystick or trackball. You've experienced the superior control. The pinpoint firing accuracy. The exceptional durability.

Features: Phoenolic ball offers 360-degree movement. Two optical encoders provide split-second movement. Quick-action fire button for smooth, two handed arcade response and feel. Long 5' computer connection. Heavy duty plastic case for long hard use.

The WICO warranty has been voided by our modification. But we give you our 15-day money back guarantee and a one-year limited warranty from Zebra Systems.

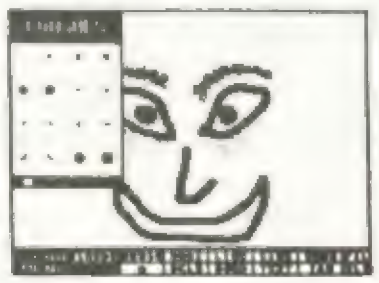
Timex Games $2 Each

With your order for a TS2068 trackball you can purchase any of the following Timex TS2068 Trackball and Joystick compatible games at the special low price of $2.00 each for cassettes and $3.00 for cartridges.

CAT#	TITLE	CAT#	TITLE
Cassettes at $2.00 each		64014	Hungry Horace
64001	Androids	64015	Horace Goes Skiing
64002	Penetrator	64019	Horace and the Spiders
64004	Casino I	64021	Blind Alley
64005	Crossfire	64023	Crazybugs
64006	Circuit Board Scramble		
64007	Dragmaster	Cartridges at 3.00 each	
64009	Guardian	74001	Androids
64012	Fun Golf	74005	Crazybugs

$5 Off Tech-Draw Jr.

You can save $5.00 on the purchase of Tech-Draw Jr. if you purchase it at the same time as a TS2068 trackball. Instead of the regular price of 19.95 you can get it for 14.95. See our catalog for a complete description of Tech-Draw Jr. and a list of printers that it supports. Order Tech-draw Jr. Catalog# C256.



CRITICAL MASS

An arcade game by Durell.
Distributed by Knighted Computers. Cat#KC12 $15.95

SABOTEUR

A martial arts arcade adventure by Durell, distributed by Knighted Computers. Cat#KC13 $15.95

MUSICOLA

Great music program for the TS2068.
New low price! Cat#TEJ1 $9.95.

GRAPHIST

Graphics software from T.E.J. Associates. New low price!
Cat#TEJ2 $9.95

SOUNDESIGN

This TS2068 programming utility allows you to design your own TS2068 sound effects and musical effects and make them a part of your BASIC programs. Wonderfully simple to use. From Arrow Software & RMG. Cat#RMG1 $14.95

CLONE

Now you can make backup copies of all of the tapes that you thought were unbreakable. New program from RMG Enterprises backs up virtually any tape that you can load into the 2068 including spectrum tapes. Includes very well written documentation package. Cat#RMG2 $9.95

THE KRUNCHER

From RMG , this utility helps you compress your BASIC programs so that you can squeeze more program into your machine. Works on TS2068 and Spectrum. Cat#RMG3 $9.95

Used ZX81's, RAM's, etc.

Working ZX81 with Zebra 15 day money back quarantee. Does not include any documentation, cables or power supply. $15.00

Not Working ZX81's (as is, usually easy to fix)	3 for $25
ZX81 Power supplies when purchased with a ZX81 (new)	$2.00
Case of 30 new ZX81 power supplies	$50 & Shipping.
Video Switch (used, working)	$1.00
Cassette Cables or Video Cable (used)	$.50 ea.
ZX81 Manual (used)	$3.00
16K RAM (used, working)	$6.00
16K RAM (not working, as is)	$2.00

BOOK BARGAINS

Here are some excellent books still in stock but not in our catalog becouse we only have limited quantities available. We'll only take phone confirmed orders on these.

Understanding Your ZX81 ROM by Logan (35 left)	#C105	$5.00
Machine Language Made Simple for TS1000 (21 left)	#C106	$5.00
The Timex Sinclair 2068 Explored by Hartnell (11 left)		$5.00

Zebra Catalog Updates

The following are typographical and other corrections to Zebra's 1986B Catalog.

Page 1 - We do not cary Omnicalc.

Page 5 - The correct cat# for Profile 2068 is TW02.

Page 6 - The last sentence of the first paragraph is in error. New A&J TS2068 drives use black, version 1 wafers.

Page 8 - The correct catalog numbers for the following cartridges are 07-7400 Pinball, 07-7300 Flight Simulator, and for cassettes 06-1000 Vu-Calc, 06-1001 Vu-File, 06-1002 Vu-3D, and 06-3000 Flight Simulator.

Page 12 - MTERM II Tape is currently priced at $24.95 not $29.95. MTERM II is not available on cartridge. We no longer sell Mini Xmod 1.7.

Page 14 - We are now sold out of 03-3020 Computer Coach, and 03-3016 Conversational Spanish.

Page 15 - We are now sold out of the following Softsync TS1000 software: SST02 Advanced Budget Manager, SST18 Mothership.

TS1000 Joystick Adaptors

Plugs into the back of the TS1000 and allows a standard Atari compatible joystick to work on the TS1000. Includes free joystick games tape. Won't work on the TS1500. This is a closeout. Only 40 left. Cat#C120 $5.00

SPRITES 2068

As you may already know, "sprites" are computer graphic objects which can be easily and quickly moved around on a video screen. Hints of how to write sprite software for the TS2068 are given in Timex's TS2068 Technical Manual but it is not really adequately explained.

Now there has been a major breakthrough. After months of research, two TS2068 dichards: Vernon Tidwell and Ron Ruegg, have now figured out how to use sprites on the TS2068. And even more importantly, they have written an excellent 34 page manual that explains all about it in their product called SPRITES 2068.

It doesn't matter whether you're a BASIC or machine language programmer - with the easy interface of the sprite service utility and the superb manual that explains it, you will be able to create your own moving sprite graphics. You may want to create complete games or just experiment and learn more about computer graphics techniques. In either case you'll enjoy the ease and the amazing high speed with which your own graphic creations will move.

SPRITES 2068 consists of the 34 page manual and a cassette tape. The cassette includes a 2520 byte machine code sprite service utility, a SPRITEDRAW BASIC program for defining and moving your own sprites (including twelve sprite commands), and exciting sprite demonstration programs.

Considering the quality of this product, its excellent documentation, the fact that there isn't anything else like it available, and most importantly what it can do for you on your TS2068, this is beauty of a program, and a bargain at only $19.00. Be sure to order yours now.

Order Catalog No. C421 SPRITES 2068 Only $19.00

ZEBRA SYSTEMS, INC.
78-06 Jamaica Avenue, Woodhaven NY 11421 (718) 296-2385

"The Mystery of the Missing 253"

Part Three

By Wes Brzozowski

A Hearty Thank You

The heading really says it all. I've been quite pleased and encouraged by your response to Part 1. This is really turning into an "interactive series", as I'd hoped, and I want to invite you to keep writing and calling with your ideas and questions. Your're truly making these articles much better than I could have done alone.

I wish I could have said all this sooner, but the publication delays on my end do get in the way. I have to submit my "stuff" about a month ahead of the publication date. The result is that I'll be submitting Part 4 about the same time you read this. In the same way, your first responses began to come just as I submitted Part 2, when it was too late for me to include a mention of them.

So you see, there's no escaping this little nuisance, and I'll just have to be content in extending a late, but very sincere thank you.

"...And Now, The Mail..."

A number of you deserve much more than just a mention for the valuable contributions you've provided. Sadly, that's all I can do. Please don't be insulted if I didn't include you here; I have to limit this much more than I'd wish.

The first pat on the back goes to Robert Orrfelt, from Redwood City, CA. He shows that you needn't use my trick to SAVE the EXROM code to tape; just put your disassembler into RAM, then type: OUT 255,128:OUT 244,16. This will switch the EXROM into chunk 4, starting at hex 8000. Really clever! If you use a Spectrum disassembler, and your emulator is in the cartridge slot (as I use), this won't work, since it would require enabling Dock and EXROM chunks simultaneously. Also, if you want to disassemble in decimal, you can't get the code to start at decimal location 4000. Still, this should be a big timesaver for almost everyone.

For reasons to be seen later, I'd like to thank Eric Johnson of Orange City, FL, and fellow SINCUS member Dave Schoenwetter for making several "dead" SCLDs available to me.

Marty Egan of Herndon, VA has also been busily studying the EXROM code, and working out Timex's bank switching protocol. I've spent a great deal of (very pleasant) time with him over the phone, as we compared out notes. I hope my infor was as helpful to you as your insights were to me, Marty. I don't just owe you one...I owe you a million.

Marty has also suggested that I include a cross-reference between a few of my terms and some of the acronym-like bank switching names that Timex included in a few spots in the Tech Manual. I chose to try to "expand" these acronyms in this series, to make the text clearer.

Timex Name	New "Improved" Name Used Here
BNA	Bank Number Access (register B0)
ABN	Assigned Bank # (A0, in setup mode)
HS	Horizontal Select (register 40)
HSP	Universal Deselect Byte (A0, in normal mode)

Timex also referred to HSP as HS-prime, but this seemed too redundant

I avoid acronyms as much as I can, and was surprised (and suitably humbled) when Rick Best, from Largo, FL asked if I couldn't include a glossary of terms in my articles; explanations of things like AROS, LROS, SCLD, ect. Well, I'll certainly be glad to explain them. (It's amazing how we can let acronyms become a part of our vocabulary without even realizing it!)

AROS (Application ROM Oriented Software) and LROS (Language ROM Oriented Software) are the two types of cartridge programs that the system can run. TM5.0 tells about these in detail. Note that AROS and LROS are "nested acronyms"; that is, one of their letters actually stands for another acronym. (A sign that these things have long since gotten out of hand. I gleefully enjoy pointing out such verbal perversities.)

The term SCLD probably stands for either Semi Custom Logic Device, or Standard Cell Logic Device, (both are true) and usually refers to the specially made "workhorse chip" inside the TS 2068. It appears that this term was intended to refer to any "special" chip to be used in TS 2068 products, and so I've also used it to refer to devices that we can only speculate about.

Another reader who's sent a large amount of infomation is William J. Pederson, owner of the Widjup Co. Mr. Pederson tells me he has a bank switching system working, which he expects to incorporate into a product. Note that some of his bank switching concepts are VERY different from what we'll be discussing here. Interested readers may wish to drop him a line to find out what's available.

If you've written me with a request for a reply, please be patient. I get swamped sometimes, and my time for writing replies is limited. Between queries on my articles in the newsletter for the SINCUS user group and now my articles here, (not to mention actually WRITING the articles) things can get very busy. But I will get to you just as soon as I can.

A Bit 'O The Hard Stuff

We talked hardware last time, but some updates may be useful. You may have noticed that it requires a huge quantity of TTL chips to implement the functions we've described. But there may be easier ways to do it. Marty Egan is investigating ways to persuade a 74LS610 chip to do some of the grunt work, and I might suggest looking at an AMD2901 bit slice chip to do the same.

Further, if we wish to rewrite the READ_BS_REG and WR_BS_REG routines, as was suggested in Part 1, a really dramatic drop in parts count seems possible. Since these routines are the only ones that actually access the bank switching hardware, they can be changed to control circuitry that's simpler to build. Since we already have to make massive bug corrections to both ROMs anyway, changing these two is trivial.

Last time, I said that the RESET signals in my block diagrams were probably not what Timex really intended, and that some odd "unlock" code was instead intended to disarm some power-on "lock up" circuitry. I'd mightily appreciate it if you'd forget I'd ever said this. (Sometimes we look at a simple problem and imagine complex solutions. Sorry, gang.) The odd code will be explained later. The reset signal really should be there, but i probably doesn't go to the backplane's RESET line.

This is because the RESET signal desn't go to a pin on the standard TS 2068 SCLD either, and so wouldn't reset the standard Horizontal Select register. If RESET only worked on an expansion bank, then applying that signal could result in some chunks not being allocated to any bank. That would hang the machine up, were it to exclude chunk 0.

Were does the signal go, then? A quick look at the sales literature for the NCR Corporation's standard cell devices (of which the 2068's SCLD is one) shows that they can include a power-on-reset circuit right on the chip. I've extracted the actual silicon chip from a dead SCLD, and sure enough, near one edge, is the large capacitor needed to perform such a function. (Well, it LOOKS large, at 500X mag.) The SCLD circuits needed to control an expansion bank probably would have had the same function inside. As such, both TS 2068 and its expansion banks would have gotten their Horizontal Select registers reset ONLY at Power-Up. That way, if an expansion bank were in control of chunk 0, and a RESET occurred, someone would still be in control.

It turns out that Chapt.5 of the "T/S 2068 Intermediate/Advanced Guide" (SAMS) has a tutorial on Extended Bank Switching, which has useful information. Unfortunately, that chapter was obviously written before the 2068's design cycle was completed, and a lot of its information has been rendered incorrect by engineering changes in the machine. it shows the old scheme, with I/O ports FC and FD as bank switching controls, making no mention of the memory mapped I/O scheme we can see in the TS 2068 code. It also makes no mention of the Universal Deselect Register, and the bank switching example given sometimes sends data out in nybbles, and sometimes as a byte.

Among the more useful gems to be found is the fact that bit 0 of a bank's status byte (bit 0 of register A0, to us) would have been set to 0, if that bank had caused an interrupt. The "Interrupt Priority", shown in the SYSCON table last time, affects the final renumbering of the banks. (High priority gives a low bank number.) This means that if we poll each bank to learn if it caused an interrupt, starting with bank #1 and working up-

ward, we will have automatically first checked the ones that demand a fast response.

As a final (and totally unrelated) hardware note, the designer should use caution in designing a Daisychain circuit. Since the clock signal is generated separately by each bank (as I showed it), the Daisychain flip flops aren't really being clocked synchronously, as is required for a shift register. This type of situation requires the use of master-slave flip flops, or two flip flops in a master-slave configuration. This will prevent one flip flop from changing its data before the next one clocks it in. If all the banks to be used are on the same circuit board however, only a single clock signal is needed, and synchronous operation is possible.

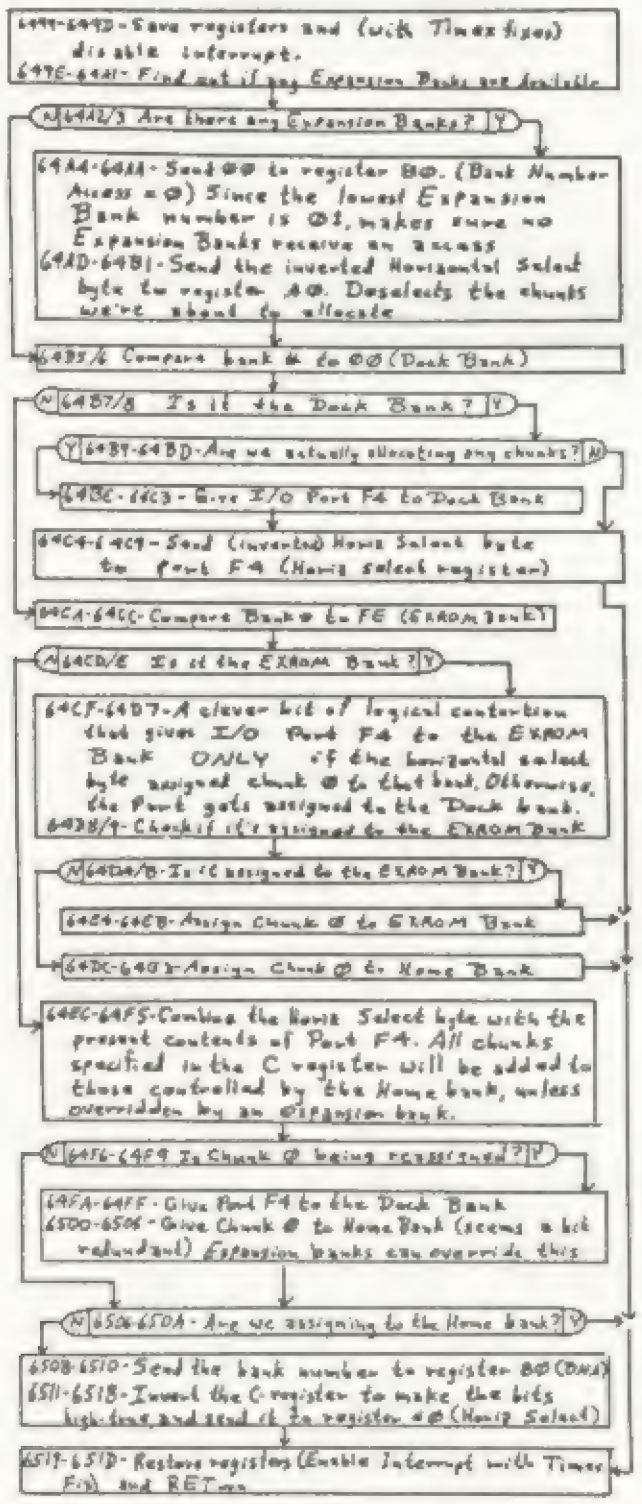
Why Bother?

This is a reasonable question. With considerable circuit complexity and ROM bugs galore, reconstructing the thing would first seem like an exercise in self-punishment. There are already simpler expansion schemes available.

As it turns out, this would be a very bad method if all we wanted was extra memory. We can now buy RAM cards that plug into the cartridge slot, and one of the available disk systems can "switch banks" that overlay one another in the Dock bank. User group newsletters have published various "RAM in the Dock slot" methods. (I published one in 1984!) But the level of 2068 software being developed today doesn't even make full use of the machine. Why would we need another way to expand it?

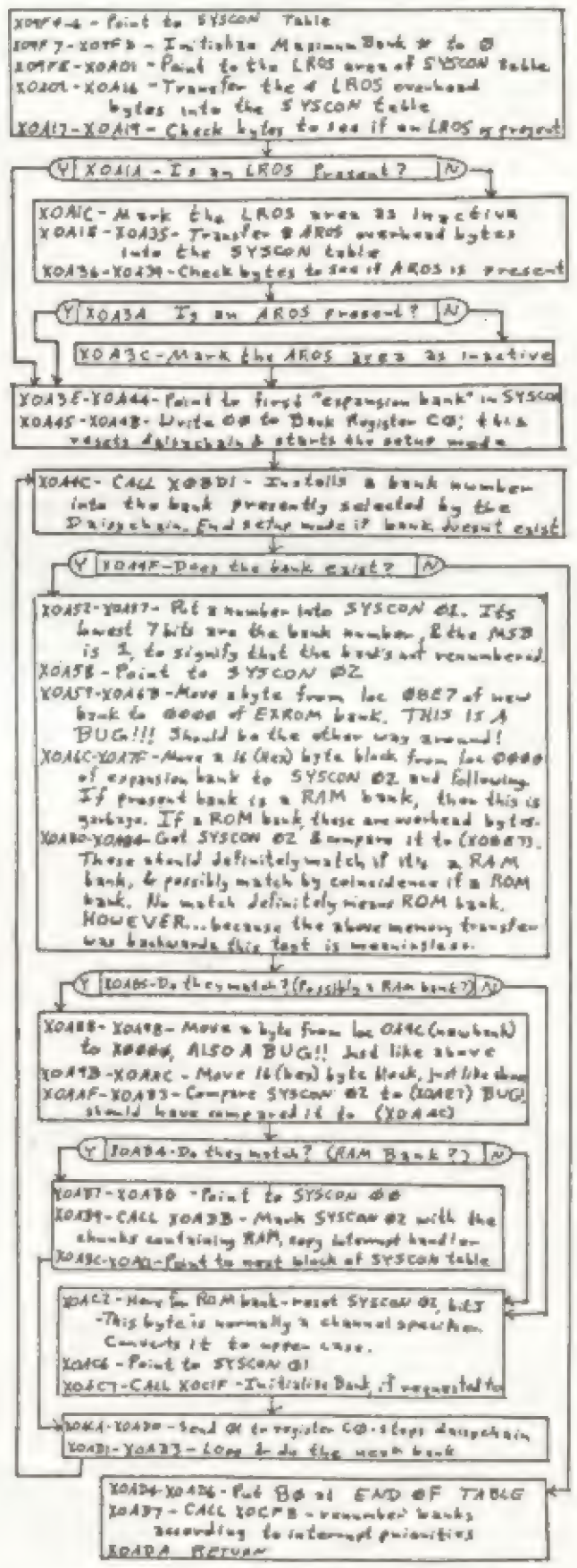
We don't simply need more memory, but we CAN use many of the undocumented (and presently bug laden) capabilities that are hidden in the ROM. If you're aware of the stream-and-channels I/O system tht the 2068 uses, you understand how it's possible to LOAD in a "print driver" program that redirects the Basic LPRINT and LLIST commands to a large printer. The 2068 tries to expand on this "Spectrum-based" theme allowing such print drivers, or any other software for an intelligent I/O device, to be located permanently in an expansion bank. These programs would take up NONE of your Home Bank memory and so wouldn't conflict with anything else running there.

But there's no reason for an I/O device to completely dominate a bank. While the extra memory space could have been taken



Flowchart 2. BANK_ENABLE in RAM Resident Code

Flowchart 3. Installs a bank number into the bank selected by the Daisychain. CY tells wether or not there's really a bank there



FLOWCHART 4. Build The SYSCON Table

up by something like an interrupt driven printer buffer, it sould also have been possible to include extra RAM, or utilities in a ROM. Further banks might have contained a disk operating system, or spiffed-up versions of the 40/64/80 column display utilities in the Technical Manual. And they could have been made directly accessable from Basic! No PEEKs, POKEs, or USR calls should have been needed.

These things just scratch the surface. The point is that the expansion banks, and some extra BEU circuitry similar in function to Sinclair's Interface One for the Spectrum, would have easily extended the TS 2068's repertoire of Basic commands to handle some very nifty I/O functions, and they'd have been immediately available when you powered up your machine. We'll begin a discussion of the 2068's I/O system and extended commands later on. Until then, keep in mind that this is where the extended bank switching system would have really made the 2068 shine!

Taking Care Of Old Business

Let's first consider Flowchart 2, which describes the BANK ENABLE routine in the RAM Resident Code. To use this, we would first put the bank number in B, and the Horizontal Select byte we want for the bank in the C register. This will work for the standard banks and expansion banks both. No one really uses it for the standard banks at the moment; it's a lot easier to program the standard banks directly. As we'll see, that's not the case if there are any expansion banks in the system.

At 64A2, we check if there are any expansion banks. If there are, we run some code to deselect the chunks specified from any expansion bank that might have them. Note that if no expansion bank has them, this can't hurt, and if we're about to give the chunks to a bank that already has them, this momentary loss won't be noticed. At 64B7, we check if it's the Dock bank we're selecting. If so, we program it directly, and we're done.

If not, we check if we're selecting the EXROM bank. If so, we pretty much do the same thing, except the code only allows us to give chunk 0 to that bank. Remember, that's the only chunk originally intended to be used there.

If it's not the EXROM bank, then it's either the home bank or an expansion bank. In either case, it doesn't hurt to try to give it to the home bank, because an expansion bank will override this if it has to. We do this at 64EC. The code from 64F6 to 6505 appears benign, but useless.

At 6506, we see if we were selecting the Home Bank. If so, then we're done. Otherwise, we send the bank number to register 80 (Bank Number Access), and the the Horizontal Select information to register 40. And that's that.

Flowchart 3 is a bit of an embarassment, because it references that incorrect "unlock" scheme I asked you to forget. (You don't remember, I hope.) My explanation will correct two errant lines in it. Since I first thought this routine controlled special hardware, it was mentioned last time. Unfortunately it doesn't, and now it would be more appropriate if I first describe the routine that CALLS it. That's the routine that builds the SYSCON table.

Daddy, Where Do SYSCONs Come From?

Well, we're mature enough in our understanding of bank switching that we know that the stork does NOT bring them! The high level initialization routine (Flowchart 1, in Part 1 of this series) CALLs the routine to build the table. Shown here in Flowchart 4, it works as follows.

We start by pointing to the SYSCON table and assuming there are no expansion banks (we'll update this assumption if and when we find some.) We then transfer the 4 LROS bytes into the SYSCON table. (TM 5.1.1 explains these bytes.) If no LROS is present, the 8 AROS overhead bytes are transferred (see TM 5.1.2). In either case, if the device wasn't present, its space is marked to show it inactive. The "bug" described in TM 6.1.4 can be corrected by having the JR at X0A1A go to X0A1E, if no LROS is present.

At X0A3E, we point to the SYSCON space for the first expansion bank and enter the setup mode. In this mode, anything written to register A0 will become the Assigned Bank Number of the bank selected by the Daisychain. Also, during the bank initialization, the HL register is always supported to point to the SYSCON location we're working with.

At X0A4C, we CALL routine that tries to install a bank number, checks to see if it succeeded, and ends the setup mode, if not. Returning from that routine, if we've run out of banks, we leave the setup loop to X0AD4, mark the end of the SYSCON table, and CALL a routine that RE-ASSIGNS the bank numbers, according to their value in SYSCON 17. This is called the Interrupt Priority.

[Editor: WOW! Wes, we ran out of space already! And just when it was getting good. We will all have to hold on to our hats 'til next issue!]

TONE/PHONE 2068? By J. Kevin Paulsen

No, this is not about modems...this is about using the TS 2068's sound chip to have a little fun. We leave it to the individual as to how enthusiastic one's fun becomes.

What we plan on doing here is simulating the tones produced by a touch tone type phone. Each button or key on a tone phone produces two tones when it is pressed. Since the 2068 has three channels of sound on the sound chip (plus another if you include the BEEP command), we easily have enough equipment to do the job.

In order to find out what tones are used I had to do some investigation. Luckily, a friend of mine at the plant where I work was taking an electronics course, and had a book at home that contained the information...and so, we're in business.

The diagram shows the layout of a standard tone phone keypad. To the left of each row of numbers is the frequency for one of the two tones produced by that number key on the phone. At the bottom of each column is the frequency for the other tone produced by that key. For example, if you press the "1" key on a phone it simultaneously produces a tone at a frequency of 697 and a tone at the frequency of 1209.

What we need to find is the coarse/fine values for the tone registers of the sound chip. On page 194 of the TS 2068 User Manual is a short program just for this purpose. A little rounding of numbers is required to get the values that come reasonably close to producing the tones we need.

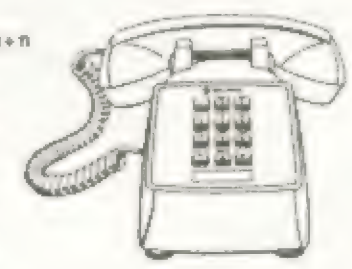
Once this is done, a short subroutine like the one in the listing can be written to simulate tone dialing. The example listing is based on the assumption that it is part of an address book type file. In this case, the file is stored in a string array--d$--whose DIMensions are something like 75 different files each 128 characters long [DIM d$(75,128)]. The phone numbers are stored starting at the 117th character in each file.

Let's review the listing: Line 2850 executes the command to open the sound chip channels, A and B, and sets up the FOR/NEXT loop for reading the phone number off the file. Lines 2852 and 2855 skip over characters which are not numerals, but are usually found in phone numbers. Line 2857 figures which line further below to call based on the number it is "reading" and calls it [GO SUB 2860+n]. Lines 2860 thru 2869 do the actual execution of the tones. The last digit of the line number corresponds to the number of the telephone key which is being simulated. Line 2870 off the tones and gives a proper break (silence) between the current and next tones. Line 2880 ends the subroutine and RETURNs you to your main program.

WARNING: You should not use this on your phone as your 2068 in not FCC approved for use as telephone dialing equipment. This is merely for simulation and fun. You certainly don't want the boys from Washington knocking on your door.

Enjoy your Tone/Phone 2068!

```
2840 PRINT "Press D to dial"
2842 GO TO 2842+(INKEY$="d")
2850 SOUND 7,60: FOR d=117 TO 128
2852 IF d$(f)(d)="-" THHEN  GO TO 2875
2855 IF d$(f)(d)=" " THEN  GO TO 2880
2857 LET n=VAL (d$(f)(d)): PRINT n;: GO SUB 2860+n
2858 GO TO 2870
2860 SOUND 0,15;9,15;0,116;1,0;2,82;3,0: RETURN
2861 SOUND 0,15;9,15;0,156;1,0;2,90;3,0: RETURN
2862 SOUND 0,15;9,15;0,156;1,0;2,82;3,0: RETURN
2863 SOUND 0,15;9,15;0,156;1,0;2,74;3,0: RETURN
2864 SOUND 0,15;9,15;0,142;1,0;2,90;3,0: RETURN
2865 SOUND 0,15;9,15;0,142;1,0;2,82;3,0: RETURN
2866 SOUND 0,15;9,15;0,142;1,0;2,74;3,0: RETURN
2867 SOUND 0,15;9,15;0,128;1,0;2,90;3,0: RETURN
2868 SOUND 0,15;9,15;0,128;1,0;2,82;3,0: RETURN
2869 SOUND 0,15;9,15;0,128;1,0;2,74;3,0: RETURN
2870 PAUSE 10: SOUND 0,0;9,0: PAUSE 1
2875 NEXT d
2880 RETURN
```



697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#
	1209	1336	1477

SIMPLY MUSIC

By S. D. Lemke

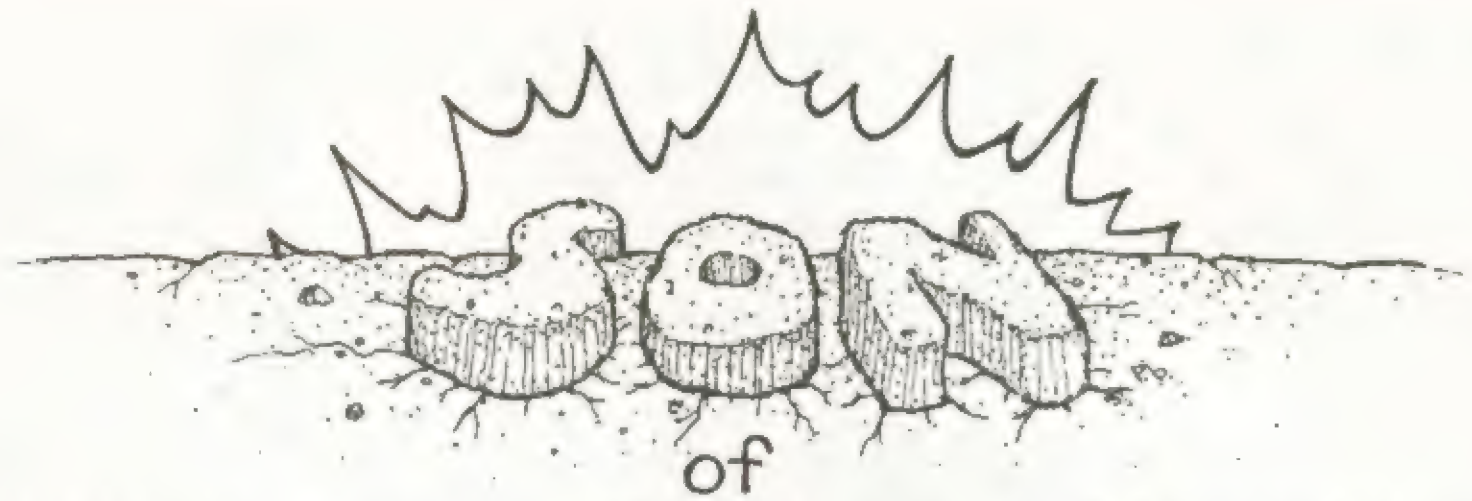
```
 10 REM ************************

        Simply  Music

 (C) by     S D Lemke
      Lemke Software Development
           2144 White Oak
       Wichita,  Ks.   67207

*********************************

  20 PAPER 1: BORDER 1: INK 7: C
LS : PRINT AT 10,7; FLASH 1;"PRE
PARING SCORE": GO SUB 8000
  30 LET p=0: LET vb=13: LET vs=
13: LET va=13: LET q=0: GO SUB 5
00
  40 LET b1=0: LET d1=0: LET f1=
0
  50 LET b=0: LET d=0: LET f=0
  60 LET b=b-1: LET d=d-1: LET f
=f-1
  70 IF INKEY$<>"" THEN  GO SUB
200
  75 IF q THEN  PRINT AT 10,15;"
?"
  76 IF NOT q THEN  PRINT AT 10,
15;"?"
  77 LET q=NOT q
  80 IF b<=0 AND p THEN )8,0
  85 IF b<=0 THEN  LET b1=b1+1:
LET b=s(1,3,b1): LET sv=vs: IF s
(1,1,b1)=0 THEN  LET sv=0
  90 IF d<=0 AND p THEN )9,0
  95 IF d<=0 THEN  LET d1=d1+1:
LET d=s(2,3,d1): LET av=va: IF s
(2,1,d1)=0 THEN  LET av=0
 100 IF f<=0 AND p THEN )10,0
 105 IF f<=0 THEN  LET f1=f1+1:
LET f=s(3,3,f1): LET bv=vb: IF s
(3,1,f1)=0 THEN  LET bv=0
 110 IF b1>n1 OR d1>n2 OR f1>n3
THEN  GO TO 400
 120)0,s(1,1,b1);1,s(1,2,b1);2,s
(2,1,d1);3,s(2,2,d1);4,s(3,1,f1)
;5,s(3,2,f1);8,sv;9,av;10,bv: GO
 TO 60
 190 REM    Adjust Settings
 200 LET i$=INKEY$: IF i$="" THE
N  RETURN
 210 IF i$="B" OR i$="b" THEN  L
ET vb=vb+1: IF vb>15 THEN  LET v
b=0
 220 IF i$="A" OR i$="a" THEN  L
ET va=va+1: IF va>15 THEN  LET v
a=0
 230 IF i$="S" OR i$="s" THEN  L
ET vs=vs+1: IF vs>15 THEN  LET v
s=0
 240 IF i$="S" OR i$="s" THEN  P
RINT AT 12,0;"SOPRANO    ";AT 12
,8;vs;AT 12,13;: FOR i=1 TO vs:
PRINT  PAPER 5;" ";: NEXT i: FOR
 i=vs TO 15: PRINT  PAPER 1;" ";
: NEXT i
 250 IF i$="A" OR i$="a" THEN  P
RINT AT 14,0;"ALTO       ";AT 14
,8;va;AT 14,13;: FOR i=1 TO va:
PRINT  PAPER 6;" ";: NEXT i: FOR
 i=va TO 15: PRINT  PAPER 1;" ";
: NEXT i
 260 IF i$="B" OR i$="b" THEN  P
RINT AT 16,0;"BASE       ";AT 16,
8;vb;AT 16,13;: FOR i=1 TO vb: P
RINT  PAPER 2;" ";: NEXT i: FOR
i=vb TO 15: PRINT  PAPER 1;" ";:
 NEXT i
 270 IF i$="P" OR i$="p" THEN  L
ET p=NOT p: PRINT AT 18,0;"Phras
ing is ";("not " AND p);"Legato.
    "
 280 RETURN
 400)8,0;9,0;10,0: PRINT #0;AT 1
,2;"Press any Key to Continue.":
 PAUSE 0: GO SUB 500: GO TO 40
 500 CLS : PRINT AT 3,8;"Simply
Music": PRINT AT 6,0;t$: LET i$=
"S": GO SUB 240: LET i$="A": GO
SUB 250: LET i$="B": GO SUB 260
 510 PRINT AT 20,0;"Press S for
SOPRANO,  A for ALTO      B for B
ASE,  P for PHRASING"
 520 PRINT AT 18,0;"Phrasing is
";("not " AND p);"Legato.      "
 530 PRINT #0; INVERSE 1;" Press
 ""ENTER"" to Start Song. "
 540 GO SUB 200: IF i$<>CHR$ 13
THEN  GO TO 540
 550 PRINT #0;AT 0,0;TAB 31;" ";
TAB 31;" ": RETURN
8000 DIM s(3,3,400):30,0;1,0;2,0
;3,0;7,56;8,0;9,0;10,0;11,50;12,
120;13,10
8001 RESTORE 8100: READ t$: READ
 n1: FOR i=1 TO n1: FOR j=1 TO 3
: READ s(1,j,i): NEXT j: NEXT i
8002 RESTORE 8110: READ n2: FOR
i=1 TO n2: FOR j=1 TO 3: READ s(
2,j,i): NEXT j: NEXT i
8003 RESTORE 8120: READ n3: FOR
i=1 TO n3: FOR j=1 TO 3: READ s(
3,j,i): NEXT j: NEXT i
8004 DATA 0,192,32,16,60,60,60,2
55,0,3,4,8,60,60,60,255
8005 RESTORE 8004: FOR i=0 TO 15
: READ b: POKE USR "a"+i,b: NEXT
 i
8006 RETURN
8098 REM

                Canon in D
                  by Pachebel

8099 REM       SOPRANO
8100 DATA "              Canon in D
                      by  Pacheb
el",56,119,1,8,245,1,8,190,1,8,8
4,2,8,51,2,8,239,2,8,51,2,8,245,
1,8,74,0,8,84,0,8,94,0,8,99,0,8,
112,0,8,125,0,8,112,0,8,99,0,8,7
4,0,8,84,0,8,94,0,8,99,0,8,112,0
,8,125,0,8,112,0,8,99,0,8
8101 DATA 74,0,8,84,0,8,94,0,8,9
9,0,8,112,0,8,125,0,8,112,0,8,99
,0,8,74,0,8,84,0,8,94,0,8,99,0,8
,112,0,8,125,0,8,112,0,8,99,0,8
8102 DATA 74,0,8,84,0,8,94,0,8,9
9,0,8,112,0,8,125,0,8,112,0,8,99
,0,8,74,0,8,84,0,8,94,0,8,99,0,8
,112,0,8,125,0,8,112,0,8,99,0,20
8109 REM       ALTO
8110 DATA 128,119,1,8,245,1,8,19
0,1,8,84,2,8,51,2,8,239,2,8,51,2
,8,245,1,8,119,1,8,245,1,8,190,1
,8,84,2,8,51,2,8,239,2,8,51,2,8,
245,1,8,94,0,8,99,0,8,112,0,8,12
5,0,8,141,0,8,149,0,8,141,0,8,16
7,0,8
8111 DATA 94,0,4,125,0,4,99,0,4,
125,0,4,112,0,4,149,0,4,125,0,4,
149,0,4,141,0,4,188,0,4,149,0,4,
188,0,4,141,0,4,188,0,4,167,0,4,
141,0,4,74,0,2,125,0,2,94,0,2,12
5,0,2,99,0,4,125,0,4,94,0,2,149,
0,2,112,0,2,149,0,2,125,0,4,149,
0,4,112,0,2,188,0,2,141,0,2,188,
0,2,149,0,4,188,0,4,112,0,2,188,
0,2,141,0,2,188,0,2,167,0,4,141,
0,4
8112 DATA 74,0,2,125,0,2,94,0,2,
125,0,2,84,0,2,125,0,2,99,0,2,12
5,0,2,94,0,2,149,0,2,112,0,2,149
,0,2,99,0,2,149,0,2,125,0,2,149,
0,2,112,0,2,188,0,2,141,0,2,188,
0,2,125,0,2,188,0,2,149,0,2,188,
0,2,112,0,2,188,0,2,141,0,2,188,
0,2,99,0,2,167,0,2,125,0,2,141,0
,2
8113 DATA 94,0,2,125,0,2,94,0,2,
125,0,2,99,0,2,125,0,2,99,0,2,12
5,0,2,112,0,2,149,0,2,112,0,2,14
9,0,2,125,0,2,149,0,2,125,0,2,14
9,0,2,141,0,2,188,0,2,141,0,2,18
8,0,2,149,0,2,188,0,2,149,0,2,18
8,0,2,141,0,2,188,0,2,141,0,2,18
8,0,2,125,0,2,167,0,2,125,0,2,14
1,0,14
8119 REM       BASE
8120 DATA 56,119,1,8,245,1,8,190
,1,8,84,2,8,51,2,8,239,2,8,51,2,
8,245,1,8,119,1,8,245,1,8,190,1,
8,84,2,8,51,2,8,239,2,8,51,2,8,2
45,1,8,119,1,8,245,1,8,190,1,8,8
4,2,8,51,2,8,239,2,8,51,2,8,245,
1,8
8121 DATA 119,1,8,245,1,8,190,1,
8,84,2,8,51,2,8,239,2,8,51,2,8,2
45,1,8,119,1,8,245,1,8,190,1,8,8
4,2,8,51,2,8,239,2,8,51,2,8,245,
1,8,119,1,8,245,1,8,190,1,8,84,2
,8,51,2,8,239,2,8,51,2,8,245,1,8
8122 DATA 119,1,8,245,1,8,190,1,
8,84,2,8,51,2,8,239,2,8,51,2,8,2
45,1,8,119,1,8,245,1,8,190,1,8,8
4,2,8,51,2,8,239,2,8,51,2,8,245,
1,8,119,1,8,245,1,8,190,1,8,84,2
,8,51,2,8,239,2,8,51,2,8,245,1,2
0
9999 SAVE "Simply M" LINE 1
```

SIMPLY MUSIC is an all Basic program that uses the three SOUND channels of the TS 2068 to create music. Each "voice" can be adjusted before and while the music is playing (although the music is interupted while the adjustment is made). Phrasing can be selected as Legato (smooth), or not smooth. A tiny metronome ticks off the beats while the music plays.

The program creates a "musical score" by READing in values from DATA statements. Each tone consists of three parts, a FINE TUNE value, a COARSE TUNE value, and the duration (in beats). See chapt.21 of the 2068 User Manual. Note durations are all relative, but in the present song, a WHOLE note gets 8 beats, a HALF note gets 4, a QUARTER note gets 2, and an EIGHTH note gets 1 beat. RESTS are input as 0 (zero). The MUSICAL SCORE begins with the DATA statement in line 8100. First is a title (in quotes). The first number is the number of notes played by this voice. Voice 1 in this case is the Soprano voice. The second number (119) is the FINE tune value, the third value (1) is the COARSE tune value, and the fourth number is the duration of the first tone, 8 beats, a whole note. The following numbers continue to define the musical score of voice one. Line 8110 starts the musical score of voice two. The first number defines the number of tones (and rests) played by this voice. This is followed by the values that define these tones. Line 8120 starts the musical score for the third voice.

This particular arrangement of "CANON IN D" starts with all three voices in harmony, and it sounds as if there is only one voice. After a few bars, the second voice appears, and a short time later, the third. Though simple, the music is effective!

Listing notes: Lines 75 and 76 each have a "?" in quotes. These are UDG "A" and "B" characters respectively. These are the tiny metronome defined in lines 8004 and 8005. Lines 80, 90, 100, 120, 400, and 8000 all have "brackets" in them. This is really the Basic SOUND command, and must be typed with the keyword SOUND. About the only way to debug this song, is to listen as it plays and seek out the "kinks". When you INPUT data from a printed score, you can actually follow the music one voice at a time and find your errors. If the program plays too slow, it can be speeded up by deleting lines 70 to 77. If you want only Legato (smooth), delete lines 80, 90, and 100 also. SAVE the program to tape after you have typed it in by "RUNing 9999". The program will auto-run when it loads. "PREPARING SCORE" will flash on the screen as the DATA is read. When completed, you will be able to adjust the voices by pressing "S" for Soprano (voice 1), "A" for Alto (voice 2), and "B" for Bass (voice 3). Press "P" to change the phrasing. Press "ENTER" to play the song.



of
Ultra-Easy Designer Graphics

by Paul Bingham

The large and warm response to the ULTRA-EASY DESIGNER GRAPHICS Program for the 2068 (which appeared in the July/August 1986 issue of Time Designs attests to the many 2068 users yearning for ways to use UDGs effectively. Many sent listings of enhancements they had added, some sent tapes, one wrote to say he had been looking for this program for a long, long time and wished I had written it sooner. 'Truth is, so do I! I think all this renewed interest in our 2068's graphic programming abilities is great.

In the first article I made mention, "that there were only 21 of them,"--UDGs that is. Well, as things turn out I was wrong again! So what appears here is some new program lines to soup up the old version 1.0 so it will do 115 UDGs at a whack instead of just 21. I call it "SON OF UDG".

Now if you crack your 2068 manual open to page 262 you will find the name CHARS listed. By reading the content note you will discover that by altering the address in CHARS we can set up an alternate table of letters and symbols in RAM and the 2068 will use them instead. How exciting! New symbols, new fonts, new graphics--its all possible. CHARS covers the Character set starting with the space (code 32) and through to the copyright symbol (code 127). This is in diference to an article on fonts I just read in SWN. The entire set is not pointed to by CHARS, only CHR$ codes 32 through 127.

Check the listing of these characters in the manual's Appendix B (page 240 and on). Now lets experiment. Type in the short Listing #1. This looks in the table in ROM and lists the values for each of the eight bytes which comprise each character. Character #124 and #126 list eight bytes the same as the rest, but the manual states they are STICK and FREE. What the table lists produces a vertical bar symbol and a reverse quote, just like the SPECTRUM. But elsewhere in the ROM, the 2068 ignors this and prints STICK or FREE...two commands the SPECTRUM does not have. Because of this fluke "SON OF UDG" ignors #124 and #126 as well, so as not to cause problems.

The program keeps track of what CHARS is set to at any given time, but in your own programs you must change the contents back to the original values before, say breaking or listing. If you don't every symbol will become total gibberish. In that case try POKEs to put things as they were: 23606 should be 0 and 23607 should be 60. In the program GOSUB 610 will perform the same service.

The "SON OF UDG" program uses all the same keys as the original plus the "a" key which is a screen toggle. One is the old graphic work slate, the other is a current list of 115 Characters (see Fig.2). You will be asked upon switching screens if you will be returning or wish the work slate's contents displayed. This is so if you toggled in mid-stream to check something that your current efforts won't be obliterated. SAVE and LOAD have also been modified to proper size for all 115.

In order to get your old listing up to "SON OF UDG" status you will need to do the following:

1. DELETE lines 10 thru 20, 36 thru 39, 43 thru 110 350 thru 520, 9010 thru 9050, 9095 thru 9120, 9220 thru 9225, line 25, and line 200
2. Alter "65368, 159" in lines 28 & 29 to "64598, 941"
3. Alter "20" in line 190 to "750"
4. Alter line 9060 by removing "PAPER 5:" command
5. Alter line 1 to include "SON OF" so you know later
6. Add all the lines of Listing #2

And thats all there is to it! You are of course welcome to make any alterations or enhancements you wish to the program (Several found grids on the work slate to be helpful last time, for example)...and feel free to send ideas and comments to me also. If you would like a complete listing of the entire "SON OF UDG" program the way it is supposed to look, just mail a dollar and I'll send you one. Write Paul Bingham, P.O. Box 2034, Mesa, AZ 85204. (If you're not up to typing, I will send copies of the complete program on tape for $5.)

Figure 1

```
120=0,0,68,40,16,40,68,0,
121=0,0,68,68,68,60,4,56,
122=0,0,124,8,16,32,124,0,
123=0,14,8,48,8,8,14,0,
124=0,8,8,8,8,8,8,0,
125=0,112,16,12,16,16,112,0,
126=0,20,40,0,0,0,0,0,
127=60,66,153,161,161,153,66,60,
```

Figure 2

```
1"-3 136 212731554156UDG:   ERASE
2#_4                 144     -1-
     502  70F  90Z  110n  145■    -2-
C#:  513  71G  91[  111o  146└    -3-
32   524  72H  92\  112p  147▒    -4-
33! 535  73I  93]  113q  148▒    all
34" 546  74J  94↑  114r  149▀    row
35# 557  75K  95_  115s  150▄   SWATH
36$ 568  76L  96£  116t  151▌   SAVE
37% 579  77M  97a  117u  152▐   CODES
38& 58   78N  98b  118v  153▀    -1-
39' 59,  790  99c  119w  154▄    -2-
40( 60<  80P100d  120x  155■    -3-
41) 61=  810101e  121y  156▌    -4-
42* 62)  82R102f  122z  157▐   all
43+ 63?  83S103g  123{  158▒   STORE
44, 64@  84T104h  124-  159▒    -1-
45- 65A  85U105i  125}  160_    -2-
46. 66B  86V106j  126~  161■    -3-
47/ 67C  87W107k  127▒  162▒    -4-
480 68D  88X108l  addr:163▒   all
491 69E  89Y109m  65464164▀   PRINT
```

Listing 1

```
  1 REM  2068 CHR$ Table Peeker
 10 FOR f=32 TO 127: PRINT f;"=
";
 20 FOR t=f*8+15360 TO f*8+1536
7
 30 PRINT PEEK t;",";
 40 NEXT t: PRINT "": NEXT f
```

Listing 2

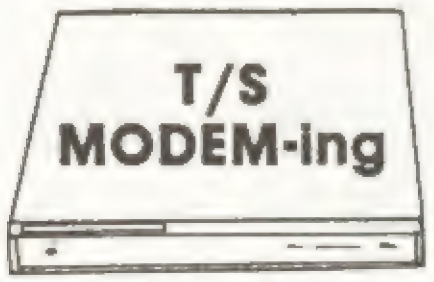
```
 10 CLEAR 64597: GO SUB 800: FO
R t=15616 TO 16384: POKE (t+4899
2),PEEK t: NEXT t: FOR t=65528 T
O 65535: READ o: POKE t,o: NEXT
t: FLASH 0: CLS : GO SUB 9015: G
O SUB 9520: DATA 0,0,63,252,252,
248,0,0
 20 DIM k(257): DIM c(4): FOR t
=1 TO 4: LET c(t)=32: NEXT t: DI
M d(32): DIM u(20): LET cs=7: LE
T at=7: LET px=5: LET py=1: INPU
T "Press ENTER to continue...";n
$: GO TO 105
 25 GO TO 350
 36 LET s=1: LET qx=0: GO TO 41
0
 37 LET s=9: LET qx=1: GO TO 41
0
 38 LET s=17: LET qx=0: GO TO 4
10
 39 LET s=25: LET qx=1: GO TO 4
10
 42 INPUT "1st:";c(1);" 2nd:";c
(2);" 3rd:";c(3);" 4th:";c(4)
 43 IF ci=9000 THEN GO SUB 350:
 GO SUB 600: PRINT AT 0,1;CHR$ c
(1);CHR$ c(2);AT 1,1;CHR$ c(3);C
HR$ c(4): GO SUB 610: FOR t=6 TO
 18 STEP 4: PRINT AT 0,t;c(1+(t-
5)/4): NEXT t: PAPER 1: RETURN
```

```
  44 FOR t=1 TO 4: IF t<=2 THEN
LET y1=1: LET x1=(INT (t*6/12)!*
8+4: GO TO 46
  45 LET y1=9: LET x1=(INT (t*3/
12)!*8+4
  46 IF c(t)>143 THEN LET h=(c(t
)-144)*8+65368: GO TO 63
  47 LET h=(c(t)-32)*8+64598
  63 FOR m=h TO h+7: LET a1=PEEK
 m
  64 FOR g=8 TO 1 STEP -1: LET a
1=a1/2: IF INT a1<a1 THEN PAPER
 0: PRINT AT y1,x1+g;CHR$ 143;: L
ET a1=INT a1: GO TO 66
  65 PAPER 7: PRINT AT y1,x1+g;C
HR$ 128;
  66 NEXT g: LET y1=y1+1: NEXT m
  NEXT t: RETURN
 100 GO SUB 9000: GO SUB 9520
 105 PAPER 1: PRINT AT cs,26;" "
: PAPER 7: PRINT AT cs,25;CHR$ 1
64: IF ci=9600 THEN GO SUB 8000:
 GO TO 110
 107 GO SUB 600: GO SUB 610
 110 PAPER 1: PRINT AT cs,26;" "
: IF CODE INKEY$=51 THEN LET cs=
cs+1: GO TO 120
 200 IF CODE INKEY$=97 THEN GO S
UB ci
 210 GO TO 105
 350 PAPER 7: IF ci=9000 THEN PR
INT AT 0,1;"  ";AT 0,6;"    ";AT
 0,10;"    ";AT 0,14;"    ";AT 0,18
;"    ";AT 1,1;"  ";AT 21,16;"
    ": RETURN
 360 FOR m=21 TO 24: GO SUB m: N
EXT m: RETURN
 400 FOR h=x TO x+7: FOR t=y TO
y+7: PRINT AT h,t;CHR$ 128: NEXT
 t: NEXT h: RETURN
 410 INPUT "CHR$ Number(C#) as S
torage:";vn: IF vn>31 AND vn<128
 AND vn<>124 AND vn<>126 OR vn=1
43 AND vn<168 THEN LET e=vn: GO
TO 420
 415 INPUT "Illegal entry!--hit
ENTER";v$: GO TO 410
 420 IF e>127 THEN LET j=INT (s/
8): LET w=e-143: LET i=65360+8*w
: GO TO 430
 425 LET j=INT (s/8): LET w=e-31
: LET i=64590+8*w
 430 LET ru=s: FOR t=i TO i+7: P
OKE t,d(ru): LET ru=ru+1: NEXT t
 435 PAPER 7: IF ci=9000 THEN GO
TO 530
 500 LET ki=i: LET f=e: FOR h=18
+j TO 21: GO SUB 600: PRINT AT h
,11;CHR$ e;AT h,23;CHR$ f: GO SU
B 610: PRINT AT h,13;ki;AT h,19;
f;: PAPER 5: BRIGHT 1: PRINT "  "
;: PAPER 7: BRIGHT 0: LET ki=ki+
8: LET f=f+1: NEXT h
 505 IF CODE CHR$ e=124 OR CODE
CHR$ e=126 THEN RETURN
 510 GO SUB 600: FOR h=19 TO 21:
 PRINT AT h,j+1;CHR$ e: NEXT h
 520 IF j<2 THEN PRINT AT 19,j+7
;CHR$ e: GO SUB 610: RETURN
 525 PRINT AT 20,j+5;CHR$ e: GO
SUB 610: RETURN
 530 IF e<50 THEN LET gx=2: LET
gy=28: GO TO 560
 535 IF e<70 THEN LET gx=6: LET
gy=48: GO TO 560
 540 IF e<90 THEN LET gx=10: LET
 gy=68: GO TO 560
 545 IF e<110 THEN LET gx=14: LE
T gy=88: GO TO 560
 550 IF e<128 THEN LET gx=19: LE
T gy=108: GO TO 560
 555 LET gx=24: LET gy=143
 560 PRINT AT 21,16;i;AT 0,6+INT
 (s/8)*4;"  ";AT 0,6+INT (s/8)*
4;e: GO SUB 600: PRINT AT gx,1+I
NT (s/17);CHR$ e;AT gy-e,gx;CHR$
 e: GO SUB 610: RETURN
 600 POKE 23606,86: POKE 23607,2
51: RETURN
 610 POKE 23606,0: POKE 23607,60
: RETURN
 785 IF ci=9600 THEN GO SUB cs+2
0: RETURN
 790 IF ci=9000 AND cs=5 OR cs=8
 OR cs>15 THEN GO SUB cs+20: RET
URN
 795 RETURN
 800 FLASH 1: PRINT AT 17,8;"jes
emenet!": RETURN
9000 GO SUB 9015: INPUT "Display
 previous work? ";n$: IF n$<>"n"
 THEN LET tx=5: LET ty=1: FOR t=
1 TO 256: GO TO 9003
9001 RETURN
9003 IF k(t)=1 THEN PAPER 0: PRI
NT AT ty,tx;CHR$ 143: PAPER 7: G
O TO 9010
9005 PRINT AT ty,tx;CHR$ 128
9010 LET tx=tx+1: IF tx>20 THEN
LET tx=5: LET ty=ty+1
9011 NEXT t: GO SUB 34  RETURN
9015 LET ci=9600: BORDER 1: PAPE
R 5: BRIGHT 1: FOR t=0 TO 21: PR
INT AT t,0;"
          ";: NEXT t
9225 PAPER 7: BRIGHT 0
9520 PAPER 1: BRIGHT 0: FOR t=0
TO 21: PRINT AT t,26;" ": NEXT t
9530 BRIGHT 1: FOR t=0 TO 21: PR
INT AT t,27;"      ": NEXT t
9540 BRIGHT 0: PRINT AT 0,27;"ER
ASE";AT 7,27;"SWATH";AT 6,27;"SA
VE ";AT 9,27;"CODES";AT 15,27;"S
TORE";AT 21,27;"PRINT"
9550 BRIGHT 1: LET a=28: LET j=1
: GO SUB 9500
9560 LET j=10: GO SUB 9500: LET
j=16: GO SUB 9500
9570 PRINT AT 6,a;"row": PAPER 7
: BRIGHT 0: RETURN
9600 INPUT "Returning to current
 work? ";n$: IF n$<>"n" THEN PAP
ER 7: GO SUB 800: LET tx=5: LET
ty=1: FOR t=1 TO 256: GO TO 9602
9601 GO TO 9605
9602 IF INT (ATTR (ty,tx)/8)<>7
THEN LET k(t)=1: GO TO 9605
9603 LET k(t)=0
9605 LET tx=tx+1: IF tx>20 THEN
LET tx=5: LET ty=ty+1
9606 NEXT t
9608 LET ci=9000: FLASH 0: BORDE
R 5: PAPER 5: BRIGHT 0: FOR t=0
TO 21: PRINT AT t,0;"
                ": NEXT t
9610 LET bb=0: LET xp=32: LET yp
=49: LET wp=28: LET xt=0: GO SUB
 9720
9620 LET xp=50: LET yp=69: LET w
p=48: LET xt=4: GO SUB 9720
9630 LET xp=70: LET yp=89: LET w
p=68: LET xt=8: GO SUB 9720
9640 LET xp=90: LET yp=99: LET w
p=88: LET xt=12: GO SUB 9720
9650 LET xp=100: LET yp=109: LET
 xt=11: GO SUB 9720
9660 LET xp=110: LET yp=127: LET
 wp=108: LET xt=16: GO SUB 9720
9670 LET xp=144: LET yp=164: LET
 wp=143: LET xt=21: LET bb=1: GO
 SUB 9720
9680 PAPER 7: PRINT AT 0,6;"
                ": PAPER 5: PRINT AT
16,25;" ";AT 16,19;"- ";AT 18,19
;"- ";AT 0,0;"1  3";AT 1,0;"2  4
"
9690 PRINT AT 0,5;"1";AT 0,9,"2"
;AT 0,13,"3";AT 0,17;"4": BRIGHT
 1: PRINT AT 3,0;"C#:";AT 0,21;"
UDG:";AT 20,16;"addr:": PAPER 7:
 BRIGHT 0: PRINT AT 21,16,"63568
";AT 0,1;"  ";AT 1,1;"  "
9700 BRIGHT 0: RETURN
9720 FOR z=xp TO yp: LET aa=z-wp
: PAPER 5: BRIGHT bb: PRINT AT a
a,xt;z;
9730 POKE 23606,86: POKE 23607,2
51: PAPER 7: BRIGHT 0: PRINT CHR
$ z: POKE 23606,0: POKE 23607,60
: NEXT z: RETURN
```

TIMEX SINCLAIR-RELATED BBS's IN THE U.S.

Courtesy Of Pete Fisher

					Access to PC Pursuit	Accepts Stacked Commands	300 BPS	1200 BPS	2400 BPS	9600 BPS or Message Base	E-Mail	20Kb Downloads	100K Downloads	Total System Capacity	Total T/S Downloads	Total CP/M Downloads	
AVERAGE REMOTE	(213)329-...1?	8,1,N	24Hrs		Y	N	Y	Y	N	1	Y	Y	Y	21 MB	1 MB	MB	Los Angeles, California
BILL'S OBSESSION	(404)377-2550	8,1,N	90%		Y	Y	Y	Y		2	Y	Y	N	22 MB	900 KB	2 MB	Atlanta, Georgia
COMPUSERVE	(Local No.)	7,1,0	24Hrs			N	Y	Y	Y	2	Y	Y	Y			Y	Columbus, Ohio
FWKUG	(817)540-4183	8,1,N	24Hrs		Y	Y	Y	Y	Y	0	Y	Y	N	44 MB	256 KB	25-30 MB	Ft.Worth, Texas
ISTUG	(317)898-3903	7,1,0	24Hrs		N	N	Y	N	N	1		N	N	64K	0	0	Indianapolis Indiana
LOONEY BIN!	(619)390-9470	7,1,0	24Hrs		N	N	Y	N	N	L	N			562 KB	0	0	San Diego, California
MCI MAIL	(Telenet)	8,1,N	24Hrs			N	Y	Y	Y	0	Y	N	N		0	0	Wash,D.C.
NIGHT OWL	(312)459-5721	8,1,N	24Hrs		Y	N	Y	Y	N	1	Y	Y	Y	7 MB	1½ MB	0	Chicago, Illinois
ONOT-NET	(718)837-2881	8,1,N	24Hrs		N		Y			1	Y	N	N		0	0	New York
ONEGO FREE ACADEMY	(607)754-3420	8,1,N	24Hrs		N	Y	Y	Y		0	Y	Y				0	Onego, New York
PLINK			24Hrs				Y	Y	Y	1	Y	N	N		0	0	
SERIAL PORT	(313)286-0145	8,1,0	24Hrs		N	N	Y	Y	Y	1	Y	Y		15 MB	20 KB		Ann Arbor, Michigan
SOURCE	(Telenet)	7,1,0	24Hrs			N	Y	Y	Y	0	Y	N	N		0	0	McLean, Virginia
STARTEXT	(817)877-1041	8,1,N	24Hrs			N	Y	Y	N	0	Y	Y	N				Ft.Worth, Texas
TIMEXCHANGE	(213)329-3922	8,1,0	24Hrs		Y	Y	Y	Y	N	1	N	Y	Y	25 MB	900 KB	20 MB	Los Angeles, California
TSU	(216)337-1099	7,1,1	24Hrs		N	N	Y	Y	N	5	Y	Y	Y	800 KB	100 KB	0	Cleveland, Ohio
VSYS	(201)527-0535	7,1,N	*		Y	N	Y	N	N	4	N	N	N		0	0	Elizabeth, New Jersey
ZEBRA SYSTEMS	(718)625-6220	8,1,N	24Hrs		N	N	Y		N	8	Y						Woodhaven New York



BASIC2text....Extending the use of MTERM II

by Michael E. Carver

If you have spent any time on a local Bulletin Board System, you will have certainly found the mighty, but humble, TS 2068 in the minority. I have had the occasion to offer BASIC programs to fulfill a few BBS users' requests. However, Sinclair BASIC is incompatible with other computer BASICs. With the help of BASIC2text, one can upload a Sinclair BASIC program via MTERM II to another brand of computer. On the receiving end, the receiver can print out the text/program and key it into their computer, making needed alterations. Some computers can compile a text file into BASIC format, allowing some receivers to simply tailor the text file to their system requirements without having to key in the program.

Sinclair BASIC is stored in the machine using many control codes (number slugs, floating point numbers, line length, ect.) and single codes for Tokens (IF, THEN, GO TO, ect.). BASIC2text will remove any non-ASCII control codes and expand the Tokens to their full ASCII equivalents.

To use BASIC2text, first LOAD a BASIC program, set RAMTOP to 64900 (CLEAR 64899), then LOAD in the machine code version of BASIC2text (LOAD "BASIC2text"CODE 64900). BASIC2text is run in two separate steps. PRINT USR 64909 will move the actual BASIC program to high memory, allowing room to build the text file. When this step is complete, a message will appear on the screen prompting you to Press Any Key to reset memory. RAMTOP will be raised to 28416, protecting the area for the text file. The screen will black out and the copyright message will appear. The machine is now ready to translate the moved BASIC into a text file (use PRINT USR 65042). The program will now convert Sinclair BASIC to an ASCII text file starting at 28416. When the translation is complete, a message will appear on the screen providing information on the start of the text file and its length. Follow the prompts to SAVE a copy to tape. IMPORTANT NOTE: Both routines must be called with the PRINT USR...not the commonly used RANDOMIZE USR. This will insure that the messages will appear on the screen.

To send the text file via MTERM II, load MTERM's buffer with the text file. I prefer to use LOADER IV. If the length of the text file is larger than the buffer area, it should be saved in two parts, allowing two smaller text files to be loaded and sent separately via MTERM.

BONUS: I have two different printer-drivers for a full-size printer, both of which do not faithfully reproduce a BASIC listing. By using the text file produced by BASIC2text, a faithful copy of the listing can be sent to a full-size printer. Set the driver's margin to 32 characters wide, and use the following BASIC program to print it to your printer:

```
10 FOR X=start of text file TO
end of text file: LPRINT CHR$ PE
EK X;: NEXT X
```

ENTERING BASIC2text

In order to save space, I have not provided a BASIC program to install the machine code. If you have access to an assembler I would suggest enterign the program via the mnemonics. If you do not have and asembler or a favorite machine code loader use Listing 2, and enter the OP Code column in the DATA statement lines.

The author can provide a copy of this program on tape for $4.00 (includes shipping). Please send a check or money order to: Michael E. Carver, 1016 NE Tillamook, Portland, OR 97212. Please specify "BASIC2text".

```
ROM CALLS
      NEW        equ 0D1D
      PR_STRING  equ 21DB
      TO_TABLE   equ 0098
      PO_SEARCH  equ 077C
      K_SCAN     equ 02B0                    Listing 1

SYSTEM VARIABLES
      PROG       equ 5C53
      VARS       equ 5C4B
      RAMTOP     equ 5C92

      ORIGIN  equ FD84  (64900d)

Address Op Code Mnemonics  Label              Notes

     ;PROGRAM VARIABLES
FD84  00        Q_FLAG      defb 00
FD85  0000      MOVED_BAS   defw 0000   ;Address of moved BASIC
FD87  0000      MOVED_END   defw 0000   ;end of moved BASIC
FD89  0000      BAS_LENG    defw 0000   ;length of BASIC
FD8B  0000      T_FILE      defw 0000   ;current pointer in Text
                                        ;file

     ;Enter here to move BASIC
     ;Call via PRINT USR 64909

FD8D  ED4B535C  MOVE_BASIC  ld bc,(PROG)
FD91  2A485C                ld hl,(VARS)
FD94  ED42                  sbc hl,bc           ;find length of
FD96  23                    inc hl              ;BASIC program
FD97  23                    inc hl
FD98  E5                    push hl
FD99  C1                    pop bc              ;Program length
FD9A  ED4389FD              ld (BAS_LENG),bc
FD9E  ED5B825C              ld de,(RAMTOP)
FDA2  ED5387FD              ld  (MOVED_END),de  ;end of moved
FDA6  2A485C                ld hl,(VARS)        ;BASIC
FDA9  EDB8                  lddr ;Move BASIC
FDAB  13                    inc de
FDAC  ED5385FD              ld (MOVED_BAS),de   ;start of moved
      ;Reset RAMTOP and NEW System              BASIC
FDB0  11DBFD                ld de,MOVED_MSG
FDB3  013A00                ld bc,003A          ;message length
FDB6  CDDB21                call PR_STRING      ;Print Message
FDB9  CDB002    WAIT        call K_SCAN         ;Wait until no
FDBC  7B                    ld a,e              ;key is pressed
FDBD  FEFF                  cp FF
FDBF  20F8                  jr nz,WAIT
FDC1  7A                    ld a,d
FDC2  FEFF                  cp FF
FDC4  20F3                  jr nz,WAIT
FDC6  CDB002    NO_KEY      call K_SCAN         ;Wait  until a
FDC9  7B                    ld a,e              ;key is pressed
FDCA  FEFF                  cp FF
FDCC  28F8                  jr z,NO_KEY
FDCE  11006F                ld de,6F00          ;New RAMTOP
FDD1  ED53825C              ld (RAMTOP),de
FDD5  CD1D0D                call NEW            ;Reset Memory
FDD8  160000    MOVED_MSG   defb 16,00,00       ;PRINT AT 0,0;
FDDB  4241534943320         defm "BASIC "
FDE1  68617320              defm "has "
FDE5  6265656E20            defm "been "
FDEA  6D6F7665642E20        defm "moved. "
FDF1  160200                defb 16,02,00       ;PRINT AT 2,0;
FDF4  505245535320          defm "PRESS "
FDFA  414E5920              defm "ANY "
FDFE  4B45592D              defm "KEY "
FE02  544F20                defm "TO "
FE05  434C45415220          defm "CLEAR "
FE0B  4D454D4F52592E        defm "MEMORY."

     ;Program to translate the moved BASIC to text file
     ;Call via PRINT USR 65042

FE12  2A825C    SET_UP      ld hl,(RAMTOP)
FE15  22B8FD                ld (T_FILE),hl
FE18  2184FD                ld hl,BFLAG         ;program flags
FE1B  3600                  ld (hl),00          ;Clear flags
FE1D  2A85FD    START       ld hl,(MOVED_BAS)
FE20  56        LINE_NO     ld d,(hl)
FE21  23                    inc hl
FE22  5E                    ld e,(hl)
FE23  23                    inc hl
FE24  23                    inc hl              ;skip length of
FE25  23                    inc hl              ;line
FE26  E5                    push hl
FE27  E5                    push hl
FE28  C1                    pop bc
FE29  2A87FD                ld hl,(MOVED_END)   ;check for end
FE2C  AF                    xor a               ;of BASIC
FE2D  ED42                  sbc hl,bc
FE2F  D2AEFE                jp nc,NOT_DONE
FE32  E1                    pop hl
FE33  1176FE    DONE        ld de,SAVE_MSG
FE36  013700                ld bc,0037          ;message length
FE39  CDDB21                call PR_STRING
FE3C  2AB8FD                ld hl,(T_FILE)
FE3F  11006F                ld de,6D84          ;Text file Start
FE42  ED52                  sbc hl,de           ;# of bytes in
FE44  2B                    dec hl              ;text file
FE45  111027                ld de,2710          ;10000d
FE48  CD69FE                call CONVERT        ;to decimal
FE4B  11E803                ld de,03E8          ;1000d
FE4E  CD69FE                call CONVERT
FE51  116400                ld de,0064          ;100d
FE54  CD69FE                call CONVERT
FE57  110A00                ld de,000A          ;10d
FE5A  CD69FE                call CONVERT
FE5D  110100                ld de,0001
FE60  CD69FE                call CONVERT
FE63  3EFD                  ld a,FD             ;lower screen
FE65  CD3012                call 1230           ;for output
FE68  C9                    ret
FE69  AF        CONVERT     xor a
FE6A  3C        COUNT       inc a
FE6B  ED52                  sbc hl,de
FE6D  3802                  jr c,PR_LENGTH
FE6F  18F9                  jr COUNT
FE71  19        PR_LENGTH   add hl,de
FE72  C62F                  add a,2F            ;obtain CHR$ code
FE74  D7                    rst 10
FE75  C9                    ret
FE76  1600      SAVE_MSG    defb 16,00,00       ;PRINT AT 0,0;
FE78  00                    nop
FE79  546F207361766520      defm "To save "
FE81  424153494320          defm "BASIC "
FE87  617320746578742D      defm "as text "
FE8F  66696C653A            defm "file:"
FE94  0D0D                  defb 0D,0D ;linefeed * ' *
FE96  5341564520            defm "SAVE "
FE9B  226E616D652220        defm ""name""
FEA2  434f444520            defm "CODE "
```

```
FEA7  323834313620          defm "28416,"
FEAD  00                    nop
FEAE  EB          NOT_DONE  ex de,hl          ;convert line #
FEAF  1E20                  ld e,20           ;to decimal
FEB1  01E803                ld bc,03E8        ;1000d
FEB4  CDB6FF                call OUT_SP_NO
FEB7  016400                ld bc,0064        ;100d
FEBA  CDB6FF                call OUT_SP_NO
FEBD  010A00                ld bc,000A        ;10d
FEC0  CDB6FF                call OUT_SP_NO
FEC3  7D                    ld a,l
FEC4  FE20                  cp 20             ;space
FEC6  2802                  jr z,STORE
FEC8  C630                  add a,30          ;obtain CHR$ code
FECA  CDC9FF      STORE     call STORE_CHAR
FECD  E1          BODY      pop hl
FECE  7E                    ld a,(hl)
FECF  23                    inc hl
FED0  E5                    push hl
FED1  FE22                  cp 22             ;Quotes
FED3  200A                  jr nz,ENTER
FED5  F5                    push af
FED6  3A84FD                ld a,(QFLAG)
FED9  EE01                  xor 01            ;Toggle Quotes
FEDB  3284FD                ld (QFLAG),a      ;flag
FEDE  F1                    pop af
FEDF  FE0D        ENTER     cp 0D             ;ENTER
FEE1  2817                  jr z,ASCII
FEE3  FE0E                  cp 0E             ;Number Slug
FEE5  2009                  jr nz,NOT_SLUG
FEE7  E1                    pop hl            ;BASIC Pointer
FEE8  010500                ld bc,0005        ;skip floating
FEEB  ED4A                  adc hl,bc         ;point number
FEED  E5                    push hl
FEEE  18DD                  jr BODY
FEF0  FE20        NOT_SLUG  cp 20             ;Control Code?
FEF2  3002                  jr nc,PRINTABLE
FEF4  18D7                  jr BODY
FEF6  FE7B        PRINTABLE cp 7B             ;ASCII?
FEF8  302A                  jr nc,NONASCII
FEFA  F5          ASCII     push af
FEFB  3A84FD                ld a,(QFLAG)
FEFE  CB8F                  res 1,a           ;Reset Token flag
FF00  3284FD                ld (QFLAG),a
FF03  F1                    pop af
FF04  FE0D                  cp 0D             ;If Enter
FF06  CC19FF                call z,UNREM      ;reset REM flag
FF09  FE3A                  cp 3A             ;If :
FF0B  CC19FF                call z,UNREM      ;reset REM flag
FF0E  CDC9FF                call STORE_CHAR
FF11  FE0D                  cp 0D             ;Enter?
FF13  20B0                  jr nz,BODY
FF15  E1                    pop hl
FF16  C320FE                jp LINE_NO
FF19  F5          UNREM     push af
FF1A  3A84FD                ld a,(QFLAG)
FF1D  CB97                  res 2,a           ;REM flag
FF1F  3284FD                ld (QFLAG),a
FF22  F1                    pop af
FF23  C9                    ret
FF24  FE80        NONASCII  cp 80             ;Block Graphics?
FF26  301D                  jr nc,BLOCK_GRAPHC
FF28  FE7C                  cp 7C             ;STICK?
FF2A  280F                  jr z,EXPAND
FF2C  FE7E                  cp 7E             ;FREE
FF2E  280B                  jr z,EXPAND
FF30  2184FD                ld hl,QFLAG
FF33  CB46                  bit 0,(hl)        ;Quotes flag?
FF35  20C3                  jr nz,ASCII
FF37  CB56                  bit 2,(hl)        ;REM flag?
FF39  20BF                  jr nz,ASCII
FF3B  D61F        EXPAND    sub 1F            ;Offset
FF3D  CD77FF                call TOKENS_1
FF40  CD6EFF                call TOKEN_FLAG
FF43  1888                  jr BODY
FF45  FE90        BLOCK_GRAPH cp 90           ;User graphics?
FF47  3004                  jr nc,GRAPHICS
FF49  3E20                  ld a,20           ;Space
FF4B  18AD                  jr ASCII
FF4D  FEA5        GRAPHICS  cp A5             ;User graphics?
FF4F  3004                  jr nc,TOKENS
FF51  D64F                  sub 4F            ;Make ASCII
FF53  18A5                  jr ASCII
FF55  FEEA        TOKENS    cp EA             ;REM
FF57  200A                  jr nz,NOT_REM
FF59  F5                    push af
FF5A  3A84FD                ld a,(QFLAG)
FF5D  CBD7                  set 2,a           ;Set REM flag
FF5F  3284FD                ld (QFLAG),a
FF62  F1                    pop af
FF63  D6A5        NOT_REM   sub A5
FF65  CD77FF                call TOKENS_1
FF68  CD6EFF                call TOKEN_FLAG
FF6B  C3CDFE                jp BODY
FF6E  3A84FD      TOKEN_FLAG ld a,(QFLAG)
FF71  CBCF                  set 1,a           ;Set Token flag
FF73  3284FD                ld (QFLAG),a
FF76  C9                    ret
FF77  117800      TOKENS_1  ld de,TO_TABLE    ;base address of
FF7A  F5                    push af           ;Token Table
FF7B  CD7C07      PO_TABLE  call PO_SEARCH    ;find Token in
FF7E  380C                  jr c,PO_EACH      ;table and store
FF80  3A84FD                ld a,(QFLAG)
FF83  CB4F                  bit 1,a           ;Token flag set?
FF85  2005                  jr nz,PO_EACH     ;Print space
FF87  3E20                  ld a,20           ;if needed
FF89  CDA5FF                call PO_SAVE
FF8C  1A          PO_EACH   ld a,(de)         ;cancel any
FF8D  E67F                  and 7F            ;inverted bits
FF8F  CDA5FF                call PO_SAVE
FF92  1A                    ld a,(de)
FF93  13                    inc de
FF94  87                    add a,a           ;if inverted
FF95  30F5                  jr nc,PO_EACH     ;end of Token
FF97  D1                    pop de            ;trailing space?
FF98  FE48                  cp 48
FF9A  2803                  jr z,PO_TRSP
FF9C  FE82                  cp 82
FF9E  D8                    ret c
FF9F  7A          PO_TRSP   ld a,d
FFA0  FE03                  cp 03
FFA2  D8                    ret c
FFA3  3E20                  ld a,20           ;Print trailing
FFA5  D5          PO_SAVE   push de           ;space
FFA6  D9                    exx
FFA7  CDC9FF                call STORE_CHAR
FFAA  D9                    exx
FFAB  D1                    pop de
FFAC  C9                    ret
FFAD  7B          OUT_SP_2  ld a,e
FFAE  FEFF                  cp FF
FFB0  200E                  jr nz,PR_DIGIT
FFB2  3E00                  ld a,00
FFB4  180A                  jr PR_DIGIT
FFB6  AF          OUT_SP_NO xor a             ;Print line #
FFB7  ED42        OUT_SP_1  sbc hl,bc         ;as decimal
FFB9  3C                    inc a
FFBA  30FB                  jr nc,OUT_SP_1
FFBC  09                    add hl,bc
FFBD  3D                    dec a
FFBE  28ED                  jr z,OUT_SP_2
FFC0  FE20        PR_DIGIT  cp 20             ;Space
FFC2  2805                  jr z,STORE_CHAR
FFC4  C630                  add a,30          ;Obtain CHR$ code
FFC6  11FF00                ld de,00FF
FFC9  E5          STORE_CHAR push hl
FFCA  2A85FD                ld hl,(T_FILE)
FFCD  77                    ld (hl),a
FFCE  23                    inc hl
FFCF  2285FD                ld (T_FILE),hl
FFD2  E1                    pop hl
FFD3  C9                    ret
```

NOTE: Code from FF77 - FFC6 has been borrowed from the ROM which handles LLIST with necessary changes for present program.

Listing 2

```
9000 CLEAR 64899: LET a$="": RES
TORE : FOR i=0 TO 73: READ d$: L
ET a$=a$+d$: NEXT i
9010 IF LEN a$()1184 THEN PRINT
FLASH 1;"Error in Machine Code D
ATA      Lines 9925-9998"''"Plea
se correct before continuing": S
TOP
9020 LET address=64900: FOR i=1
TO LEN a$-1 STEP 2
9030 POKE address+INT ((i-1)/2),
(CODE a$(i)-(48 AND CODE a$(i)<5
8)-(55 AND CODE a$(i)>64))*16+CO
DE a$(i+1)-(48 AND CODE a$(i+1)<
58)-(55 AND CODE a$(i+1)>64)
9040 NEXT i
9050 CLS : PRINT "Machine Code h
as been Loaded    into memory."'
'"Press any key to SAVE & VERIFY
  BASIC2text": PAUSE 0: SAVE "BA
SIC2text"CODE 64900,592: CLS : P
RINT "Rewind and play tape to Ve
rify": VERIFY "BASIC2text"CODE 6
4900,592
9925 DATA "0000000000000000"
9926 DATA "00ED4B535C2A4B5C"
9927 DATA "ED422323E5C1ED43"
     . . .Continue this pattern
     using Line numbers 9928-9997 in
     increments of 1. . .
9998 DATA "FD7723228BFDE1C9"
```

FREE MONITOR! and I.C.E.

YES! IT'S TRUE!! Just order a QL Computer and a QL printer from VARIETY SALES and you will get, absolutely FREE, a Magnavox (or equivalent brand name) Monochrome Monitor along with the proper cable and everything else you need to start a small business or home computer set-up:

A+ Computer Response Authorized Dealer

===> Sinclair QL with 90 day warranty
===> Magnavox (or equivalent) monochrome monitor
===> Sinclair QL Near Letter Quality Printer
===> Psion Quill full-feature Word Processor
===> Psion Archive full-feature Data Base
===> Psion Easel Business Graphing/Charting
===> Psion Abacus Electronic Spreadsheet
===> All cables, manuals, wallet of 4 microdrive cartridges, power supply, ribbon for printer

And you get it all for just: $599

Send check or Money Order for $599
+$12 shipping to:

VARIETY SALES
325 WEST JERSEY STREET #2D
ELIZABETH, NJ 07202
PHONE: 201-527-0535

*** For TOTAL Quantum Leap SUPPORT ***
*** YOU Need QUANTUM COMPUTING'S ***
*** QL DISCOUNT SUPPLY CATALOG! ***
*** Get Specialized Programs! ***

QL USERS: STEP UP TO THE PERSONAL MAINFRAME SPEED THAT'S INSIDE THE QL!
OVER 1.4 MEGABYTES of FLOPPY STORAGE in Place of Mdv1 & Mdv2!
LOAD QUILL & DOCUMENTS IN UNDER 4 SECONDS!!!

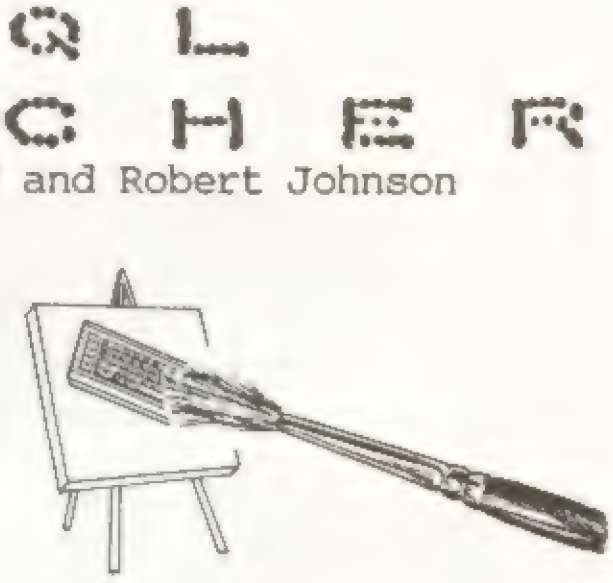
Our Quantum Buying Power enables us to offer You BRAND NEW 5 1/4 DUAL VERTICAL HALF HEIGHT FLOPPY DRIVES IN A CASE WITH POWER SUPPLY & CABLE. Fully Assembled and PRETESTED WITH A QL, READY for you to Simply Hook Up and GO! Inquire about Item **#H04. The Price is now reduced over 30% to $277.00!!** It's also availabe with Full Height Drives for Much Much LESS! Call us up on our **24 HOUR Q LINE** for Instant Information! Just another one of the Incredible Quantum Computing Bargins we always make available to our Fellow QL Users & Admirers. Get our **QL USER CATALOG** for ALL of them! **NOTE: FREEWARE with ALL 1st Purchases!** Just Enclose A BLANK Microdrive Cartridge. We will Return it to you FULL!!! ALL Disk Drive Orders will get the FREEWARE on a FREE Floppy Disk! Get Size_Base, Maps, Clocks, Machine Code Graphics Demo, File Master, etc.! **Don't Miss Out** On This!!

Order From: QUANTUM COMPUTING, Box 1280, Dover, NJ 07801
To CALL The Q Line Dial: (201) 328-8846

*** You Must ACT NOW!!! ***
*** It's a Limited Intro Offer! ***

QL ETCHER

By David and Robert Johnson



```
1 MODE 256: CLS:PAPER 0:PAPER #0;0:INK 7:CLS #0:PRINT"           SINCLAIR QL"
2 AT 3,0:PRINT "-----------------------------------":PRINT:PRINT:PRINT"     THIS
 IS AN ETCH-A-SKETCH"
3 PRINT"  PROGRAM CREATED FOR THE STRANGE"
4 PRINT" AND  DEMENTED; POSSIBLY FOR THOSE"
11 PRINT"   WE BELIEVE  THAT THIS PROGRAM "
12 PRINT"  WILL PROVIDE  MINUTES OF ENJOY-"
13 PRINT"  MENT, SECONDS OF ECSTACY, AND A"
14 PRINT" BETTER  OUTLOOK ON LIFE, THE UNI-"
15 PRINT"      VERSE, AND EVERYTHING."
16 PRINT"           ( TV MODE )":PRINT:PRINT
17 FOR K=1 TO 35
18 PRINT "-";"";
19 NEXT K
20 PAUSE 100
22 PRINT"DO YOU WISH TO SEE THE DIRECTIONS";,"(Y/N)":INPUT DI$
23 IF DI$="N" THEN GO TO 40
24 CLS:CLS #0:PRINT"    DIRECTIONS ARE AS FOLLOVS:"
25 PRINT:PRINT:PRINT"                USE:"
26 PRINT:PRINT "0-FOR UP"
27 PRINT:PRINT "L-FOR DOWN"
28 PRINT:PRINT "P-FOR RIGHT"
29 PRINT:PRINT "O-FOR LEFT"
30 PRINT:PRINT "Q-FOR DRAWING COMMANDS"
35 PRINT:PRINT "T-FOR TEXT COMMANDS"
40 PAUSE 300: CLS:CLS #0
42 PRINT "TO BEGIN"
43 PRINT:PRINT"1) DRAWING"
44 PRINT:PRINT"2) SEE A PICTURE"
45 PRINT:PRINT"PRESS 1 OR 2":INPUT YES$
46 IF YES$="1" THEN GO TO 349
47 IF YES$="2" THEN GO TO 150
48 GO TO 40
120 IF E$="Y" THEN GO TO 150
130 IF E$="N" THEN GO TO 322
140 GO TO 110
150 LET D$ = CHR$(93):LET L$ = CHR$(93)
160 PRINT D$;"OPEN POINTS1"
170 PRINT D$;"READ POINTS1"
180 LBYTES mdv1_POINTS1,131072
210 PRINT D$;"CLOSE POINTS1"
211 PAUSE 100
349 CLS:MODE 512
350 PAPER #0;7:INK #0;0:CLS #0: PRINT #0;"0=UP,
    L=DN, O=←, P=→, Q=STOP FOR COMMANDS"
370 LET Y=1
390 LET X=Y
395 LET Y$=INKEY$
400 IF Y$="" THEN GO TO 395
420 IF Y$="0" THEN LET Y=Y+1
425 IF Y$="L" THEN LET Y=Y-1
430 IF Y$="O" THEN LET X=X-1
435 IF Y$="P" THEN LET X=X+1
436 IF Y$="Q" THEN GO TO 560
437 IF Y$="T" THEN TEXT
520 POINT X,Y
521 PRINT #0;"X=";X;" Y=";Y
522 INK 7
550 GO TO 395
560 CLS #0:PRINT #0;"DO YOU WISH TO PLACE A CIRCLE
    AT X";X;"Y";Y;" ?  (Y,N)":INPUT #0; C$
```


Program Continued On Page 36...

```
565 IF C$="Y" THEN GO TO 570
566 IF C$="N" THEN GO TO 600
570 PRINT #0;"CIRCLE PARAMETERS: RADIUS, ECCENTRICITY, ANGLE":INPUT #0; Z,W,V
575 CIRCLE X;Y;Z,W,V:PRINT #0;"DO YOU WISH TO FILL ? (Y/N)"
576 INPUT #0;C$:IF C$="Y" THEN PAINT
577 IF C$="N" THEN GO TO 395
600 PRINT #0:PRINT #0;"DO YOU WISH TO CHANGE X & Y ?  (Y/N)"
601 INPUT #0; C$:IF C$="Y" THEN GO TO 610
602 IF C$="N" THEN GO TO 395
610 PRINT #0;"YOU ARE NOW AT X";X;"Y";Y;" INPUT NEW X":INPUT #0; X
611 PRINT #0;"NEW Y ??":INPUT #0; Y:GO TO 395
13000 DEFine PROCedure PAINT
13010 PRINT #0;" WHAT COLOR ? ( 0 - 7 )":INPUT #0;COLOR
13015 INK COLOR:FILL 1:CIRCLE X;Y;Z,W,V:FILL 0
13020 END DEFine
13999 REMark ************************************
14000 REMark BY David Johnson and Robert Johnson
14500 REMark for the Sinclair QL : 1986
14510 REMark ************************************
15000 DEFine PROCedure TEXT
15010 PRINT #0;"TEXT AT CURRENT X ";X;"Y ";Y;" POSITION ?? (Y/N)"
15015 INPUT #0;TEX$:IF TEX$="Y" THEN GO TO 15017
15016 IF TEX$="N" THEN GO TO 600
15017 PRINT #0;"INK ? (0-7)":INPUT #0;COLOR:INK COLOR
15030 PRINT #0;"INPUT YOUR TEXT: ":INPUT #0;TE$
15035 CURSOR X,Y:PRINT TE$
15040 END DEFine
```

Program Continued From Page 34

HOW TO USE THIS PROGRAM

This is a drawing program for the QL, and is self-explanatory--with directions in the program itself. You can view the program on a TV (F2) or a monitor on F2. The most important thing is to have your CAPS LOCK ON! Just follow the programs built in prompts.

Anyone who wants a copy of this program on Microdrive, just send a formatted cartridge to: D. Johnson, 2399 St. Rd. 95, Edison, OH 43320. Include $1.00 for shipping.

"I Built A QL Kit"

By Bob Howard, WA6DLI

As soon as you read the title, you will say.."He did WHAT". Yes, its true...like the early Sinclairs, you can now buy a QL Kit by mail order from A+ COMPUTER RESPONSE in Keene, New Hampshire (and some QL dealers also have them now).

Well, as I am primarily a 2068 buff...I ordered one as a way to dabble in the QL without a major investment. The kit price is $139 plus $7 shipping from A+. I ordered the QL kit on September 25, and it arrived October 8th.

But you say... isn't this a dirty trick...selling the QL by mail as a $139 kit when they are also expecting the 17 or so QL dealers to sell the assembled QL package at $299 list less what the dealers want to throw in as discounts or added software and accessories? I think not as you have to consider what you DON'T get with the kit. First you will be in the true Atari ST or Amiga "class" as your computer kit comes with absolutely NO SOFTWARE! This is more of a problem than you think, as you can't run to your local downtown store and buy some. Also while you can buy commercial software from the QL dealers...they don't offer the four bundled programs that come with the dealer-sold QL (word processor, data base, spread sheet, and graph programs). Since you didn't get the programs...you don't get the standard QL documentation either. (The QL kit only comes with parts of the User Guide, and there is no nice binder either. Most of the documentation concerns technical aspects of the QL.) The kit sales might hurt dealer sales of complete QL packages, but you could look at it as an expanded market for the dealers sale of peripherals and software.

If you don't need the business package...then the QL kit is a great "deal", for learning SuperBASIC and for typing in programs from tutorials such as in ZX Computing Monthly from England and also TDM.

Well enough said on the ethics of the deal...what is the QL kit like and how hard is it to build? First of all, a kit it is not...it is a knocked down QL out of the case and consists of: a case with keyboard in the top half, two micro-drives, a heat sink, and a single mother-board computer, assembled and apparently tested by A+ (derived from the stickers on the bottom of the case.) Also a bag of screws and miscellaneous parts like covers for the ports not used. The heavy power supply and cord, and TV switch box and lead is also packed in

the box. Also supplied are two blank micro cartridges, and two cables; an RGB cable and a serial port cable. (Note: these last two items are not included with pre-assembled QL's and are an option.)

The "Kit" is about as difficult to assemble as the average lawn chair or knocked down hardware you might get at a department store. This doesn't mean there are not pitfalls (you might be a klutz at reading the directions!). The QL Kit comes with the following documentation: An assembly manual produced by A+ Computer Response, A Beginners Guide to the QL by Sinclair, A Technical Description of the QL by Sinclair. It is all well packaged and the instructions are very good as far as they go. They look like they might have been produced with a QL graphics program.

I had the thing together in no time at all...but I am an old hand at stuffing the Sinclair keyboard ribbons into those slots! This is the most difficult part, along with not dislodging it all when you are plugging the LED wires into their sockets. These wires and the keyboard ribbons are both coming from the top lid of the case and you must do a balancing act to hold the lid at a 45 degree angle while you push the wires into sockets. One slip and you risk ruining the ribbons or may crumple the ribbons while fighting to get the LED wires into their holes and held in until you push down on the socket to lock the wires in place.

The wires for the two Microdrives can only go in the right way if you don't twist them and you have the drives in the right position as shown in the drawings.

My big trauma came when everything worked fine (keyboard all keys, LED lights, and drives)...but I couldn't pass the formatting test. Then I read the QL Beginners Guide and found that the formatting command: FORMAT MDV1_ shown in the assembly manual must include the underline symbol as part of the command (or you get the dreaded "not found" report). I thought the "_" was just their way of indicating the following flashing cursor! So my microdrives were OK but my command was incomplete. This needs to be stated in the assembly manual I think...at least it cost me a lot of grief. Oh yes, the TV switch box and cable allowed me to test the QL on a convenient TV set nearby.

A+ Computer Response does offer a phone consultation service for kit problems from 3 to 5 pm Eastern time. I am sure this is for kit assembly and test procedures only...they will not be willing to hold your hand on how to use SuperBASIC and otherwise program your QL...and your phone bill couldn't stand this either.

When you move from that TV set, you are going to find that Sinclair expected you to purchase the Sinclair RGB Monitor. You won't be able to use the QL's monitor mode on most TV's, but you could on a green or amber hi-res monitor if you know how to connect one up. The QL's RGB plug (an 8-pin DIN plug) is a rare bird to buy...its not at Radio Shack. I happen to have color monitors in my computer room/ham shack and one is a TI composite and the other is a Comrex CR-6600 RGB. Fortunately, I had been through the RGB cable/plug mess in getting my 2068 onto the RGB monitor.

What is my verdict on the kit? I feel that if you want a "bargain" in a "super computer" (with the understanding of the hassels you will have to go through to hook it up to bargain monitors, ect.), then the QL Kit is a good deal, especially if you want to program in SuperBASIC or other languages. If you want to use ICE (a GEM like desk top format operating system) and the bundled business software, you would be ahead to buy an assembled QL from an authorized dealer. You will be buying other software and peripherals from them anyway, so you might as well get off to a good start by getting the computer from them too.

How do I like my QL? Well, it is great, and I have had fun trying some of the QL programs in ZX Computing. Now...if I just could get color on one of my monitors!

For further information on the QL Kit, contact A+ Computer Response, 69-B Island St., Keene, NH 03431 (603-357-1800).

QL Quill/Word Processor Tips

Part I
by
Mike de Sosa

QL Word Processor AKA QLWP AKA Quill--the least acclaimed of the four Psion software programs bundled with the Sinclair QL--is still a good word processor, especially with added memory and RAMdisk. Quill's chief fault is that it is a bit slow in carrying out some operations. Quill's chief virtue is its ease of use; it is even simpler than Tasword II for the TS 2068. So much for criticism, now for some tips.

In this and future articles on Quill, I will first deal with rather elementary things which it is essential for any user of Quill to master and then with more complex matters.

If you have not already done so, clone a working copy of Quill from the master Quill cartridge.

Put a blank or no longer needed Microdrive cartridge in Microdrive 1. If it is a new cartridge, format it five times using:

```
FOR F=1 TO 5: FORMAT MDV1_
```

Otherwise, put your master Quill program cartridge in Microdrive 2, then key and enter:

```
LRUN MDV2_CLONE_BAS
```

This will take about ten minutes. When complete, return your Quill master program to its protective case and store it in a safe place, load a formatted file cartridge in Microdrive 2, then key and enter:

```
LRUN MDV1_BOOT
```

Quill should load in under 20 seconds. You are now ready to write! (To load and run Quill from boot_up, just insert a Qull program cartridge in Microdrive 1 and key F1.)

Quill like most software programs has preset (or default) values for line spacing, margins, tab settings, ect., so you may, if you wish, proceed immediately. (To set or check what values are set you will have to use various commands.) If you are not impatient to begin the great American novel, hold off a few minutes, and let's check out your Quill monitor screen.

At the top is the control area where prompts and reminders are shown and where additional instructions will appear from time to time. For HELP it says to press F1. Try it. Once in the HELP facility, key F1 again for instructions on how to use the facility. Key ESC to return to the program.

Keying F2 "toggles" the control area on and off, creating a larger working area (you can usually infer what's going on without the control area visible by referring to the status area--the three lines below the working area.)

Reading to the right in the control area is a block indicating that you can move the red cursor using the cursor (arrow) keys. (You cannot move the cursor on a blank screen or beyond the end of the text for the first time using the cursor keys; if you wish to leave a space at the top of the working area or later between paragraphs, you must use the ENTER key which starts a new indented paragraph or the SPACE bar or TABULATE key. Using ENTER to do this has the disadvantage of creating a new paragraph each time it is keyed which will slow your later movement through the text using the SHIFT and up and down cursor keys.)

With text on the screen, keying the up and down cursor keys moves the cursor up or down one line; keying the left and right cursor keys moves the cursor one character space left or right. Depressing the SHIFT key while keying the up and down cursor keys moves the cursor up or down one paragraph at a time. Depressing the SHIFT key while keying the right and left cursor keys moves the cursor right or left a word at a time.

Type in a paragraph of four or five lines; DO NOT USE THE ENTER KEY TO CHANGE LINES--just keep on typing without regard to where you are on a line and don't attempt to separate words at the end of a line or correct any errors. Quill will change lines for you. Now key ENTER to begin a new indented paragraph. Type a two or three line paragraph, then key ENTER again to begin a third indented paragraph. Practice moving the cursor right and left and up and down using the cursor keys and SHIFT plus the cursor keys. Do not worry that you cannot always place the cursor precisely where you wish: this is an unfortunate quirk of Quill! Check "Cursor" in the HELP facility.

The wide central window in the control area displays the information shown upon loading Quill, two sets of commands when F3 is keyed, and screen prompts during command sequences. The top line of the center window indicates you are in the Insert mode wherein characters keyed appear at the cursor position, displacing any existing text to the right--note that if more than one word is inserted the text will separate to permit a longer section of text to be inserted. Contrary to what it says in you QL User Guide (QLUG), the text will not rejoin itself automatically. To rejoin the text, place the cursor one space past the final character at the front of the separation and press CTRL and the right cursor key.

The bottom line in the central window of the control area advises how to change to the Overwrite mode, the other Quill mode, by depressing SHIFT and keying F4. In the Overwrite mode, which you will find is much slower than the Insert mode, you can type over existing tex. Use of the Overwrite mode, which I tend to forget is available, is frequently quicker and more useful way to edit text. Note that the current Quill mode is indicated in the status area. Check "Insert" in the HELP facility.

The second item in the central window of the control area reminds you to key ENTER to begin a new indented paragraph. Check "ENTER key" in HELP.

The third line indicates that to delete text, you depress CTRL and a cursor key. CTRL and the left cursor key delete the character to the left of the cursor. CTRL and the right cursor key delete the character under the cursor; CTRL and the cursor key delete all text on the line to the left of the cursor; CTRL and the right cursor key delete all text on the line under and to the right of the cursor. Depressing the SHIFT and CTRL keys and the left cursor key deletes the word to the left of the cursor; SHIFT, CTRL, and the right cursor key delete the word to right of the cursor. Check "Delete" in HELP.

The window to the right of the central window in the control area reminds you to key F4 to select another of Quill's other four typefaces (bold, underlined, high [superscript], and low [subscript]. Combinations are possible, for example, bold, underlined, high script. Another option is made available by keying F4--the Paint option with which the typeface of existing text may be changed; again, combinations are possible. Key F4 and follow screen prompts to add bold and underlined text, superscripts, and subscripts to your practice paragraphs. Use the Paint option to change the typeface of existing text. Check "Typeface" in HELP.

The upper right window in the control area prompts you to key F3 to select and toggle between two sets of Quill commands.

In Quill, unlike Archive, the command to be selected must appear in the central window of the control area. Once a command sequence is selected, subsequent prompts and instructions will appear in this window. A command is selected by keying the first letter of the command. Key F3, then Key F3 again, noting the commands available. When the command Justify is displayed, Key J. Use the up cursor key to move the cursor to the beginning of the second paragraph. Press the SPACE bar and note that the justification of the text in the second two paragraphs is changed. Note also that text cannot be added while in a command sequence. Key ENTER to return to the normal (Insert or Overwrite mode.) It is not a good idea to use ESC to terminate a command sequence; in some cases this might cancel a desired command change.

ESC is used to abort a command sequence in progress or to perform some designated function within a command sequence.

The working area consists of 17 lines of text with the control area present or 21 lines without the control area.

The status area consists of the three lines at the bottom of the screen. The uppermost of these is the input line editor on which the cursor, command sequence in use, and prompts sometimes appear, and on which entries (filenames, ect.) are made. The cursor will appear on this line when an input is required. The Quill mode, typeface, number of words typed, document name, and the page and line number of the cursor line are displayed on the bottom two lines in the status area.

PRACTICE, PRACTICE, PRACTICE all of these procedures now or you may develop ingrained bad habits which will slow you down later!

So far, this article has dealt with elementary but essential procedures that must be mastered. For the novice, don't wait for the next issue of TDM to continue learning Quill. Make use of the Quill section of the QLUG, and the HELP facility to teach yourself to use the program. Make sure you fully understand each command sequence as you proceed. If you are using the basic 128k QL, I advise you to SAVE your document every twenty or thirty minutes on two Microdrive cartridges and begin a new document file when the document can no longer be stored in RAM, that is, when Microdrive 2 begins to operate during text insertion. Leave at least 30 sectors free on your file cartridge when creating longer documents. Next time out I will assume that you have progressed to "the more experienced Quill user" status.

Tips for the More Experienced Quill User

Once your program is configured using CONFIG_BAS and your printer data is installed using INSTALL_BAS, delete these programs and INSTALL_DAT from your Quill working copy to make room for auxiliary SuperBASIC and machine code programs relevant to word processing. On an unexpanded QL, it is a toss-up wether you should add machine code programs to multitask with Quill; it may be better to save most of the unused RAM for document files. But you can expand your BOOT file to include many auxiliary procedures and functions without reducing the available RAM for document files significantly.

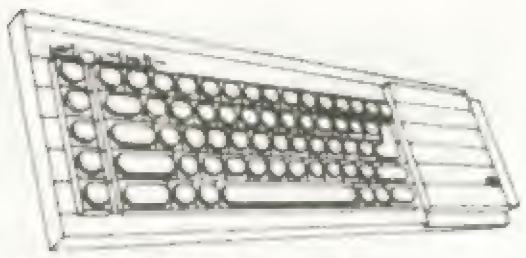
On my 640k QL, I multitask four programs with Quill: QDTG, a date-time-group program which appears in the status area, based on a program appearing in QUANTA and three proprietary programs, CAPS, QUILL_KEY and MINI_CALC. My BOOT program proper consists of about 46 lines and uses QL TOOLKIT II commands. The bulk of the BOOT program consists of about 25 defined procedures and functions. With Quill loaded in RAMdisk, I can quickly QUIT Quill, perform any necessary tasks--most frequently saving my current document file to Microdrives--and return to Quill in a flash.

Listing 1 is my Quill BOOT program. It can be easily modified to suit your needs and equipment mainly by deleting lines. Listing 2 is a machine-code program loader for a program, QtoRAM1, which transfers Quill from Microdrive 1 to RAMdisk 1, making necessary provisions for efficient RAM management. Listing 3 is the QDTG program loader.

Most of the defined procedures and functions in Listing 1 are, I trust, self-explanatory. If you can't figure something out, drop me a line, in care of TDM and include a self-addressed, stamped envelope.

[If you would like me to send you my Quill BOOT program on Microdrive, including non-proprietary machine code programs; the programs in listings 2 and 3; four PRINTER_DAT programs; and a few useful SuperBASIC programs of my own devising, you may order it through TIME DESIGNS for $15. Send check/money order to: TDM, 29722 Hult Rd. Colton, OR 97017. VISA and MASTERCARD charges accepted--telephone orders O.K. Please specify "Mike's Quill Cartridge" when ordering.]

NEXT MONTH: More on Quill, particularly on increasing the number of Quill typefaces readily available to you.



Listing 1

```
2 FORMAT ram2_200: SBYTES ram2_space,131072
,60000
4 CLEAR: WINDOW 512,256,0,0: CSIZE 1,1: CLS
6 PRINT "  ram1 = ";
8 FORMAT ram1_240
10 PRINT "  ram5 = ";
12 FORMAT ram5_360
14 PRINT " Setting up QUILL on RAMdisk"
16 COPY mdv1_quil_hob TO ram1_quil_hob
18 COPY mdv1_compare_exe TO ram1_compare_ex
e
20 COPY mdv1_solhead2_doc TO ram5_solhead2_
doc
22 COPY mdv1_lhead_doc TO ram5_lhead_doc
24 COPY mdv1_acad_doc TO ram5_acad_doc
26 COPY mdv1_fastcopy TO ram1_fastcopy
28 PRINT: PRINT " Do you wish to set clock
(Y/N)? "
30 IF INKEY$(-1)=="y": PRINT :PRINT " SDATE
 yyyy,mm,dd,hh,mm,ss"\\" Key and ENTER  'C'
 to continue": STOP
32 CLS: PRINT " Executing multitasked progr
ams"
34 EXEC mdv1_quill_key
36 EXEC mdv1_mini_calc
38 EXEC mdv1_caps
40 EXEC mdv1_qdtg
42 CLS: PRINT " Transferring Quill to RAM1_
"
44 EXEC_W mdv1_qtoram1
46 CLS: PRINT "  Select Printer Driver"
48 PRINT "  1 - Std STAR SG-10"
50 PRINT "  2 - Std STAR Delta 10"
52 PRINT "  3 - Std EPSON FX80 Compatibles"
54 PRINT "  4 - Book Manuscript"
56 INPUT " Your choice? ";pr
58 SELect ON pr
60 =1: COPY mdv1_printer1_dat TO ram1_print
er_dat
62 =2: COPY mdv1_printer2_dat TO ram1_print
er_dat
64 =3: COPY mdv1_printer3_dat TO ram1_print
er_dat
66 =4: COPY mdv1_printer4_dat TO ram1_print
er_dat
68 END SELect
70 CLS: PRINT " Copy MDV2 files to RAM5"
72 WCOPY mdv2_,ram5_
74 PRINT "More? ": IF INKEY$(-1)=="y": GO T
O 72
76 FORMAT ram2_
78 PRINT " Executing QUILL"
80 CLOSE #1: CLOSE #2: WINDOW #0,400,20,35,
215
82 EXEC_W ram1_quill
84 OPEN #1,con: OPEN #2,con
86 wscr
88 CSIZE 1,1: PRINT " Copy ram5_ document f
iles to mdv2_"
90 WCOPY ram5_,mdv2_
92 CLS: DIR mdv2_: PAUSE 150
94 PRINT: PRINT " Key and enter 'reb' to re
boot QUILL"\\"   or 'lreb' to load more doc
uments and reboot QUILL"\\"   or 'cop 1' /
'cop 2' to backup files on mdv1_ or mdv2_"
96 STOP
9000 REMark PROCEDURES & FUNCTIONS
9002 DEFine PROCedure C
9004   CONTINUE
9006 END DEFine
9008 DEFine FuNction SGN(n): IF n=0: RETurn
 0: ELSE RETurn n/ABS(n)
9010 DEFine FuNction F2C(f): RETurn (f-32)*
5/9
9012 DEFine FuNction C2F(C): RETurn C*9/5+3
2
9014 DEFine FuNction R10: RETurn RND(1 TO 1
0)
9016 DEFine FuNction R100: RETurn RND(1 TO
100)
9018 DEFine FuNction DICE: LOCal a,b: a=RND
(1 TO 6): b=RND(1 TO 6): RETurn a+b
9020 DEFine PROCedure LIST1154
9022   LOCal a,b,n$
9024   CLS #2
9026   INPUT " Program name? ";n$
9028   INPUT " Enter program start line ";a
9030   INPUT " Enter program end line ";b
9032   OPEN #3,ser1
9033   PRINT #3,CHR$(27);CHR$(82);CHR$(6)
9034   PRINT #3,CHR$(27);CHR$(77);CHR$(11)
9036   PRINT #3,CHR$(27);CHR$(81);CHR$(54)
9037   PRINT #3,CHR$(27);CHR$(78);CHR$(6)
9038   PRINT #3,CHR$(14);n$: PRINT #3
9040   PRINT #3,CHR$(27);CHR$(66);CHR$(4)
9042   LIST #3, a TO b
9044   PRINT #3,CHR$(27);CHR$(66);CHR$(5)
9046   PRINT #3,CHR$(12)
9048   CLOSE #3
9050 END DEFine
9052 DEFine PROCedure wscr
9054   WINDOW #0,508,40,4,216: WINDOW 508,2
16,4,0: WINDOW #2,508,216,4,0
9056   PAPER 0: INK 7: PAPER #2,0: INK #2,4
9058   MODE 4
9060   CLS #0: CLS: CLS #2
9062 END DEFine
9064 DEFine PROCedure DSCR2
9066   WINDOW#0,480,56,16,200: WINDOW#1,180
,200,320,0: WINDOW#2,295,200,16,0
```

```
9068   PAPER#0,0: INK#0,7: CLS#0: PAPER 0:
INK 7: CLS: PAPER#2,0: INK#2,7: CLS#2
9070   MODE 4
9072 END DEFine
9074 DEFine PROCedure DSCR
9076   WINDOW#0,480,56,16,200: WINDOW#1,480
,200,16,0: WINDOW#2,480,200,16,0
9078   PAPER#0,0: INK#0,7: CLS#0: PAPER 0:
INK 7: CLS: PAPER#2,0: INK#2,7: CLS#2
9080   MODE 4
9082 END DEFine
9084 DEFine PROCedure listdir
9086   CLS: OPEN #3,ser1
9088   OPEN_NEW #4, ram5_MF2
9090   PRINT #3,CHR$(27)&"R"&CHR$(10)
9092   PRINT #3,CHR$(27)&"N"&CHR$(6)
9094   PRINT #4,CHR$(27)&"N"&CHR$(6)
9096   PRINT #3,CHR$(27)&"M"&CHR$(10)
9098   PRINT #4,CHR$(27)&"M"&CHR$(10)
9100   INPUT "MDV Number? ";k$
9102   PRINT "q to quit"
9104   INPUT "MDV Name? ";f$
9106   IF f$=="q" THEN GO TO 9122
9108   PRINT #3,CHR$(27)&"E"
9110   PRINT #3,CHR$(14);f$: PRINT #3
9112   PRINT #4,CHR$(14);f$: PRINT #4
9114   DIR #3,"mdv" & k$ & "_"
9116   DIR #4, "mdv" & k$ & "_"
9118   PRINT #3
9120   PRINT #4: GO TO 9104
9122   CLOSE #3
9124   CLOSE #4
9126 END DEFine
9128 DEFine PROCedure reb
9130   GO TO 80
9132 END DEFine
9134 DEFine PROCedure LREB
9136   CLS: PRINT "Select MDV2 documents to
 RAM5"
9138   WCOPY mdv2_,ram5_
9140   PRINT "More? ":  IF INKEY$(-1)=="y":
 GO TO 9138
9142   GO TO 80
9144 END DEFine
9146 DEFine PROCedure cz
9148   CLS#0
9150 END DEFine
9152 DEFine PROCedure VAL
9154   LOCal y,f$
9156   FORMAT ram8_10
9158   CLS: PRINT "Input formulae?   (z to
end)"
9160   CLEAR
9162   INPUT f$;
9164   IF f$=="z": END DEFine
9166   OPEN_NEW #4, ram8_work
9168   PRINT #4,"9174 y = "&f$
9170   CLOSE #4
9172   MERGE ram8_work
9174   REMark working space
9176   DELETE ram8_work
9178   PRINT " = ";y
9180   GO TO 9160
9182 END DEFine
9184 DEFine PROCedure COP(n)
9186   WCOPY ram5_,"mdv"&n&"_"
9188   DIR "mdv"&n&"_"
9190 END DEFine
9192 DEFine FuNction root (number,root): RE
Turn number^(1/root)
9194 DEFine FuNction fact(n): IF n=1: RETur
n 1: ELSE RETurn n*fact(n-1)
9196 DEFine PROCedure sample
9198   LOCal ans%
9200   CLS
9202   INPUT "Percent pro or for candidate
A? ";a
9204   b=100-a: PRINT
9206   INPUT "Size of sample? ";n
9208   ans%=1.96*SQRT(a*b/n)
9210   PRINT
9212   PRINT "Sampling error is plus or min
us ";ans%;" percent (at 95% confidence leve
l)"\\
9214   PRINT "Range pro or for candidate A
= "; a-ans%;" to "; a+ans%;" percent"\\
9216   PRINT "Range con or for candidate B
= "; b-ans%;" to "; b+ans%;" percent"\\
9218   PRINT "NOTE: Non-sampling errors may
 exceed the sampling error!"\\
9220   PRINT "Expand ranges plus/minus 2-4%
 for greater confidence factor."
9222 END DEFine
9224 DEFine PROCedure QL2
9226   wide=254
9228   WINDOW 250,206,254,0: WINDOW #2,wide
,206,2,0: WINDOW #0,2*wide,50,254-wide,206
9230   PAPER 0: INK 4: BORDER 1,7,0,3: PAPE
R #2,0: INK #2,7: BORDER #2,1,7,0,3: PAPER
#0,0: INK #0,4
9232   FOR f=0,1,2: CLS#f
9234 END DEFine
9236 DEFine PROCedure sav(drive,name$)
9238   DELETE "mdv"&drive&"_"&name$
9240   SAVE "mdv"&drive&"_"&name$
9242   DIR "mdv"&drive&"_"
9244 END DEFine
9246 DEFine PROCedure QLS
9248   WINDOW#0,512,50,0,206: INK#0,4:PAPER
#0,0:WINDOW 256,206,257,0:PAPER 2:INK 7:BOR
DER 1,255:WINDOW#2,256,206,0,0:PAPER#2,7:IN
K#2,0:BORDER#2,1,255
9250   CLS#0:CLS:CLS#2
9252 END DEFine
```

Listing 2

```
1 REMark QtoRAM1 Loader
2 REMark Courtesy Barry Ashfield in QUANTA
4 RESTORE 14
5 start=RESPR(1024):checksum=0
6 FOR f=start TO start+279
7 READ byte:POKE f,byte
8 checksum=checksum+byte
9 NEXT f
10 IF checksum<>21753: PRINT "error in data
": STOP
11 DELETE mdv1_qtoram1
12 SEXEC mdv1_qtoram1,start,280,256
13 PRINT "QtoRAM1 saved ok": STOP
14 DATA 96,14,0,0,0,0,74,251,0,6
15 DATA 81,95,114,97,109,49,112,11,114,255
16 DATA 116,127,78,65,65,250,0,208,112,1
17 DATA 114,255,118,1,78,66,74,128,103,4
18 DATA 96,0,0,172,73,250,0,216,40,136
19 DATA 67,250,0,214,112,71,116,14,118,255
20 DATA 78,67,74,128,103,4,96,0,0,146
21 DATA 67,250,0,194,34,17,112,24,116,255
22 DATA 78,65,74,128,103,4,96,0,0,126
23 DATA 73,250,0,166,40,136,34,72,118,255
24 DATA 32,122,0,160,112,72,75,250,0,158
25 DATA 36,21,78,67,74,128,103,4,96,0
26 DATA 0,94,112,2,78,66,65,250,0,118
27 DATA 112,1,114,255,118,2,78,66,74,128
28 DATA 103,4,96,0,0,70,73,250,0,114
29 DATA 40,136,112,73,75,250,0,110,36,21
30 DATA 118,255,34,122,0,94,78,67,74,128
31 DATA 103,4,96,0,0,40,67,250,0,88
32 DATA 112,70,78,67,74,128,103,4,96,0
33 DATA 0,24,112,2,78,66,32,122,0,60
34 DATA 112,25,78,65,74,57,0,2,128,238
35 DATA 102,248,96,12,32,124,0,1,0,1
36 DATA 52,120,0,204,78,146,114,255,112,5
37 DATA 118,0,78,65,0,10,109,100,118,49
38 DATA 95,113,117,105,108,108,0,10,114,97
39 DATA 109,49,95,113,117,105,108,108,0,0
40 DATA 0,0,0,0,0,0,0,0,0,0
41 DATA 0,0,0,0,0,0,0,0,0,0
```

Listing 3

```
100 REMark QDTG Loader
105 REMark Courtesy Derek Wilson in QUANTA
110 C=RESPR(100)
120 FOR i=0 TO 68 STEP 2
130 READ x: POKE_W i+C,x
140 END FOR i
150 SEXEC mdv2_QDTG,C,100,256
1000 DATA 29439,29697,28683,20033,17402
1010 DATA 48,13944,200,20115,12040
1020 DATA 28691,20033,17402,74,-27698
1030 DATA 13944,236,20115,8279,-11314
1040 DATA 13944,208,20115,16961,16962
1050 DATA 30463,28688,20035,24794
1060 DATA 0,7,240,10,168,246
```

curry computer

— SPECTRUM SOFTWARE —

DLAN	$14.95	Zombie	$14.95	
Rebel Star Raider	$5.00	Green Berets	$15.95	
Make A Chip	$5.95	Snowman	$9.95	
Dynamite Dan	$11.95	Great Space Race	$12.95	
They Sold A Million	$18.95	Rambo	$14.95	
The Planets	$18.95	Quazatron	$16.95	
Delta Wing	$9.95	One Man And His Droid	$9.95	
Chiller	$9.95	Skyranger	$12.95	
ACE	$17.95	Bombjack	$14.95	
Cauldron II	$16.95	Jack The Nipper	$14.95	
NOW Games	$16.95	Max Headroom	$16.95	
Paper Boy	$14.95	Dynamite Dan II	$14.95	

And More-Write for Free Catalog!

******PACKAGE SPECIALS******

Buy 1 ROM Switch at Regular Price and get 2 FREE Magazines OR 1 FREE Program of Our Choice.

Buy 3 of the $3 T/S 1000 SW for $8

Buy 3 of the $2 T/S 1000 S/W for $5

Buy 3 of the $6 T/S 2068 S/W FOR $15

Buy 3 of the $2 T/S 2068 S/W for $5

— T/S 2068 SOFTWARE —

Xadom	$6.00	Bugaboo	$6.00	Spelling	$2.00
Mined Out	$6.00	Timegate	$6.00	States & Caps	$2.00
Saboteur	$15.95	Fighter Pilot	$15.95	Horace & The Spiders	$2.00
Tasword Keyboard Overlay	$5.95	Mscript Keyboard Overlay	$5.95	Fun Golf	$2.00
ROM Switch	$45.00	ACZ General Ledger	$24.95	Stock Market Sim.	$2.00
3D Strategy	$6.00	Speed King Joystick	$17.95	And More - Write for Free Catalog	

— T/S 1000 SOFTWARE —

2040 Paper for 3 rolls	$6.95	Red Alert	$3.00	3D Grand Prix	$2.00
Munchees	$3.00	Croaka-Crawler	$3.00	Vegas/Jeopardy	$2.00
Pioneer Trail	$3.00	T/S Destroyer	$3.00	ZXDB (as is)	$2.00
Stock Market Calculator	$3.00	Alpha Vowels	$3.00	Graphic Golf	$3.00
Rocket Man	$3.00	ZXtricator	$3.00	Forty Niner	$3.00
Carpooler	$2.00	Kasino Kraps	$2.00	Tarot	$2.00
Backgammon	$2.00	The Gambler	$2.00	And More - Write for Free Catalog	

— BOOKS AND MAGAZINES —

Hackers Handbook	$11.95	The Sinclair Story	$12.95	Sinclair User	$4.00
Guide to the Hobbit	$6.95	Putting Your Spectum to Work	$5.95	Computer & Video Games	$4.00
Beginners Guide to FORTH	$6.95	Mastering M.C. on the ZX81	$8.95	Your Computer	$4.00
Creat. Arcade Games/Spectrum	$5.95	Delving Deeper / Spectrum	$7.95	QL World	$4.00
49 Explosive Games on the ZX81	$6.95	Spectrum M.C. Made Easy II	$10.95	3 mags shipped at one time	3/$10.95
Replicating Reality	$12.95	Database Primer	$10.95	(some back issues available.)	
Creat. Adventures on Spectrum	$7.95	Using a Modem W/Your Computer	$12.95		
60 Programs for Spectrum	$12.95	101 Things to do W/Dead Compt	$5.95	And More - Write for Free Catalog	

* FREE S/H on all Software. Add .75 on Books if ordering alone - FREE S/H if ordering with Software.
* Add $1 S/H on ROM Switch, Speed King Joystick and on 2040 Paper Packs.
* All Software or Software/Book orders over $50 are shipped via UPS 2nd Day Air FREE (Cont. U.S. Only.)
* Please Write or Call for a FREE Catalog for Timex/Spectrum; QL or Atari ST Software, Hardware and Peripherials.
* Toll Free ORDER Line: 1-800-628-2828 Ext. 950 24 hrs. a day, 7 days a week.
* Call for Info on our "VIDEO CATALOG" of software.

P.O. Box 5607 • Glendale, Arizona 85312-5607 • 1-602-978-2902 • Telex (via WUI): 6501267701

OFFICE: 2412 West Greenway • Suite B-10 • Phoenix, Arizona

curry computer

THE PC TALKER

An exciting new Hardware item for the QL. Plugs into the Ser2 port. Utilizing some very special speech chips TALKER can say practically anything you type, and the "voice" is very understandable.

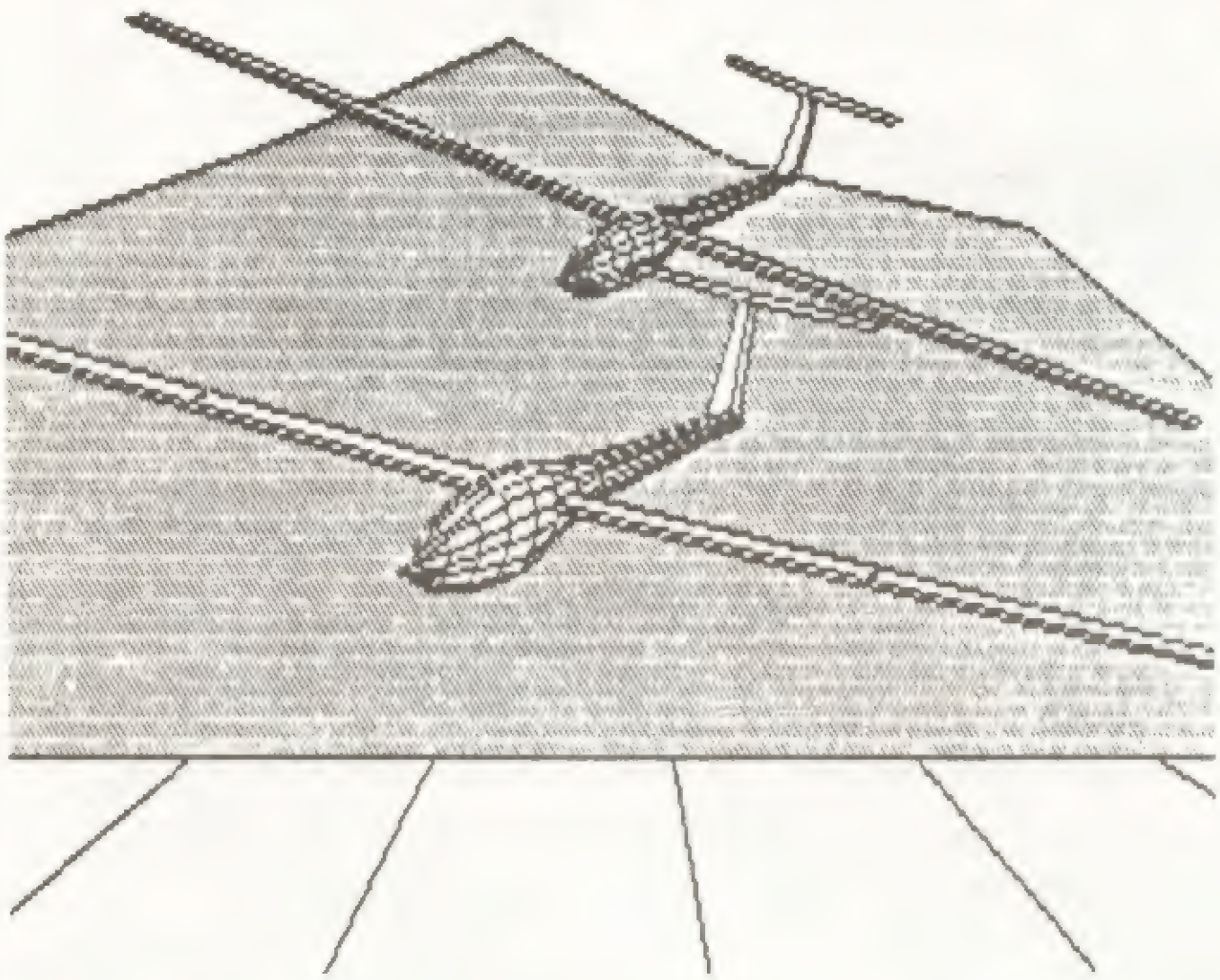
Extremely easy to program using PRINT statement. A demo program is included when you purchase TALKER.

TALKER is professionally housed in black ABS plastic. It comes with a 90 day warranty and includes its own power supply.

PC TALKER is $64.95 plus $2.95 s/h.

CONCEPT 3D

A completely interactive program. All functions and choices are called with single key entry. All required information and prompts are displayed on screen when required. We like to think of the program as a "3D Graphics Processor" since many of the functions are much like those of a good word processor.



"The United States has small part of the QL pie but, if quality products such as Concept 3D continue to come from the country which owns Silicon Valley, the American software houses concerned should congratulate themselves."

Sinclair User
Oct. 1986

Call our Toll Free ORDER Line to order either of these fine products.
1-800-628-2828 ext. 950 24 hrs. a day - 7 days a week. Or call1-602-978-2902 for more information on our catalog.

We know the QL...We've been appreciating it longer.

curry computer

- QL GAMES -

The Lost Pharaoh	$24.95	Baron Rouge	$16.95	Karate	$24.95
Dragonhold	$29.95	Vroom	$24.95	Hyperdrive	$19.95
Othello	$24.95	Early Learning	$24.95	Flight Simulator	$29.95
Mortville's Manor	$34.95	Sqadrons	$29.95	BJ in 3D Land	$19.95
Presidents	$14.95	3D Slime	$19.95	Bounder	$24.95
Wanderer	$34.95	QL Chess	$34.95	Knight Flight	$24.95

- QL GRAPHICS -

GraphiQL	$34.95
TenchniQL	$69.95
Concept 3D	$39.95
QL Pientre	$29.95
Graphic Toolkit	$29.95

- QL UTILITIES -

Screen Dump (Epson Compat.)	$15.95
Choice Cartridge (use w ICE)	$22.95
I.C.E. ROM Cartridge	$34.95
RAM Disk Software	$24.95
QCode - Terminal S/W	$29.95
Cartridge Doctor	$23.95

- MISC. PROGRAMS -

QSpell	$29.95
Assembler Workbench	$39.95
QL Mail Merge	$24.95
Nucleon	$34.95
Superchecking	$14.95
Cosmos	$24.95

- QL HARDWARE AND ACCESS. -

Thompson Case holds 12 Carts.	$2.50	Talker - Speech Synthesizer	$64.95	Monochrome Cable	$14.95
Avatex Modem 300/1200	$109.95	RS232 Cable	$14.95	RGB Cable	$16.95
512K RAM Expansion	CALL	Modem Cable	$15.95	Cumana Disk I/F	CALL

- QL BOOKS AND MAGAZINES -

Adv. Programming on the QL	$12.95	Introduction to SuperBASIC	$12.95	QL Computing	$9.95
Word Processing on the QL	$12.95	Desktop Computing on the QL	$12.95	QL SuperBASIC	$11.95
Using Graphics on the QL	$12.95	Profiting from the QL	$12.95	QL Report per Year...	$15.00
Machine Code Prog. on the QL	$12.95	Database Mgmt. on the QL	$12.95	QL World	$4.00
Making the Most of the QL	$12.95	Quantum Theory	$11.95	Sinclair User	$4.00
QL Games Master	$12.95	QL Adventure Handbook	$9.95	Your Computer	$4.00

- FOR BRITISH QLs ONLY -

Backgammon	$24.95	Early Learning	$24.95	Flight Navigation	$24.95
Mortville's Manor	$34.95	CAD Pak	$24.95	Joystick Adapter	$14.95

FREE Shipping on all Software. Add .75 per Book if ordering alone - FREE S/H if ordering with Software
Add $2.95 for Modem and Talker; $1 for Cables.

All Software of Software/Book orders over $50 are shipped via UPS 2nd Day Air FREE (Cont. U.S. Only)

Please Write or Call for a FREE Catalog for QL; Timex/Spectrum or Atari Software, Hardware and Peripherals.

Toll Free ORDER Line: 1-800-628-2828 ext. 950 24 hrs a day - 7 days a week.

P.O. Box 5607 • Glendale, Arizona 85312-5607 • 1-602-978-2902 • Telex (via WUI): 6501267701
OFFICE: 2412 West Greenway • Suite B-10 • Phoenix, Arizona

CLASSIFIED

FREE ADS FOR SUBSCRIBERS

HOW WILL TAX REFORM AFFECT YOU? Calculate 87, 88, 89 tax. Use current or future data. Include itemizing, not business. 2068 tape. $5 ppd. Max Schoenfeld, 2612 Princeton, Cleveland, OH 44118.

MONEY MACHINE II starring BANNA BRITE! All new format. Banna turns the letters. The Ultimate Word Game for TS 2068 from ABBA SOFT. $12.00 postpaid. Herb Bowers, 2588 Woodshire Circle, Chesapeake, VA 23323.

WANTED: INTERCOMPUTER QSAVE fast loading system for TS 1000. Also any TS 1000 hardware working or not. Want to swap software? Send me your list. Dan Elliott, Rt 1, Box 117, Cabool, MO 65689.

FOR SALE: RAMEX MILLENIA K DDI. Retail- $200.00 - call with offer. Also, Tasman C CPI - retail $70 - call with best offer. Bill Nemitz 412 W. Golf Ave., Ottumwa, Iowa 52501. (515) 683-1367.

WANTED: HUNTER BOARD and/or other TS 100 "what have you's". Contact: Fred Henn, c/o Rockelman & Henn Pump Co., Inc., 1333 Military Rd., Buffalo, NY 14217.

CLOSEOUT!!! TS 1000 SOFTWARE from just 99¢. Box 2382, La Jolla, CA 92038.

2068 SOFTWARE: Send S.A.S.E. for FREE CATALOG to: TIMEWARE, INC., 1907 1/2 W. Genesee St., Syracuse, NY 13204.

FOR SALE: QL, Sinclair Monitor and Printer. Memory boards, software, new blank cartridges, various QL books, QL technical manual, spare power supplies, ect. Call for prices and details. (603) 847-3448.

FOR SALE: SEIKOSHA SP-1000A NLQ Dot Matrix Printer. 2 Months old. $130 ppd. UPS. Jerry McKouen, 2580 Lanier Dr., Lansing, MI 48911.

FOR SALE: TIMEX SINCLAIR 1000 Computer in METAL SLOPING CABINET, with keyboard, power switch, Westridge Modem, Aerco Centronics Parallel Interface, Memotech 64k RAM, RS cassette recorder, large library of software including: QSAVE, Extended Basic, Sincartist, Memotext WP, Games. Available as complete package or by piece. If interested, please contact: Richard Beier, 1 Darwin Dr., N. Merrick, NY 11566. Leave EMAIL on CompuServe U.I.D. #73137, 1565.

SINCUS NEWS- $8/year= 6 issues. Heavy on 2068 help, hints and programs: Mail check to SINCUS, 1229 Rhodes Rd., Johnson City, NY 13790- a non profit, all volunteer user group for 1000/2068 users.

NEW BOOK 1000/2068 of 20 (mostly) Basic program listings and explanations: Grocerylist, Edit-Writer (wordprocessor), Treasure-tract (accounting), Danceshoes, Minutes, Songs (sing-a-long), Studygame, Barg (graph-maker) and more. Start where I left off! Send $25 Canadian to: BLUE VIOLET PUBLISHING Inc., 1452 Kingsdale, Gloucester, Ont., K1T-1G9 Canada, for a copy.

SOFTWARE FOR THE TS 2068. Send SASE for FREE CATALOG & price list to: E. Ray Rash, 2424 SW 78th St., Oklahoma City, OK 73159

Do you have some equipment or a program that you would like to sell? Looking for something hard to find? Place an ad in **THE CLASSIFIEDS!** Subscribers can place one free personal ad in each issue. Ad size is 32 Col. wide (like 2040 paper) and maximum of six lines. For additional lines - $3 each. **NON-SUBSCRIBERS** and **DEALERS:** $4 a line. **DEADLINE FOR ALL CLASSIFIED ADS:** Two weeks before publication date. Mail your ad to: **TIME DESIGNS MAGAZINE, The Classifieds Dept., 29722 Hult Rd., Colton, Oregon 97017.**

NEW PRODUCTS

For TS2068 and SPECTRUM

Artworx

VERSION 1.1

- Pull-Down Menus
- Several Brushes
- Spray
- Auto-Fill
- Zoom
- Undo
- Several Text Fonts
- Cut & Paste Windows
- Auto-Speed Control
- Magnify & Reduce
- Rotate & Mirror
- Full Attribute Control
- Fully Elastic Shapes including Circle, Box, Triangle, Ray and Line
- Fast Ellipse and Arc
- Includes Spectrum & TS2068 Versions
- Supports Microdrives and Kempston Joystick
- Includes GALLERY, the slide show/animator
- 5 Samples of Artwork
- Excellent Manual

Unshackle your creativity with ... ARTWORX!!

ARTWORX V1.1 - $19.95 U.S. plus $3.00 S & H

REVOLUTIONARY NEW BASIC COMPILER...

TIMACHINE

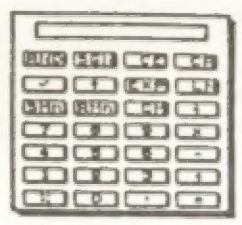
The dream of every BASIC programmer has now been realized!

* TIMACHINE will turn your BASIC into super-fast machine code, running up to 200 times faster!
* Handles floating point operations like SIN, COS, TAN
* Handles all BASIC except I/O
* Includes an excellent manual and 4 demonstration programs
* Compiles up to 27K in seconds
* Includes Spectrum & 2068 versions

Super-Charge your BASIC programs with...*TIMACHINE!*

TIMACHINE — $19.95 U.S. Plus $3.00 S & H

THE



WORX!

A unique combination of planning aids, decision tools, and utilities.

* Scheduler/Planner
* Notepad
* Telephone Book
* Programmable Calculator
* Decision Factoring
* Real Time Clock
* Biorhythms
* Perpetual Calendar
* International Time Zones
* Superb Manual
* Includes Spectrum & TS2068 Versions
* Supports Microdrives

A TIMELY ADDITION TO YOUR SOFTWARE LIBRARY...THE WORX!

— $19.95 U.S. Plus $3.00 S & H

NovelSoft

A FORMAT FOR THE FUTURE

106 Seventh Street, Toronto, Ontario, Canada M8V 3B4 • TEL. (416)259-8682 • CompuServe 70416,1435

WEYMIL presents...

A small collection of truly innovative products for Sinclair computers

THRUST TS1000
True hi-res graphics for the TS 1000. THRUST is two great programs on one tape. Sinc-Artist 1.3 is the most user friendly software ever developed for printer graphics. Completely cursor and menu driven. For Hunter Board users, there's Sinc-Artist HR which produces fantastic screen graphics. THRUST is both the most sophisticated and easiest to use hi-res graphics software ever developed for the TS 1000. If you are ready for no-nonsense, hassle free graphics, then THRUST is a "must have."

THRUST for the TS 1000 only $20.00

ARTWORX V1.1 TS 2068
England's finest graphics package! ARTWORX V1.1 establishes a new standard for color graphics with features never before available on small systems. Auto speed control, pull down menus, unique cut and past windows, ZOOM!, elastic shapes, multiple fonts, CENTRONICS I/F capability for full-size print-outs, and more. All this plus an absolutely uncluttered screen for full creativity. Easy to use. The joystick controls EVERYHTING except text entry. The highly supportive well-written documentation is almost unnessesary.

ARTWORX V1.1 for the TS2068 only $19.95

PIXEL SKETCH AND GRAPHICS EDITOR V2.0 TS 2068
This program by Stan Lemke still remains the finest graphics program produced in the United States and one of the best in the world for the TS 2068. It has had excellent user group reviews and is a snap to use. Well written, step-by-step documentation guides you effortlessly from loading to producing you own "masterpiece." Great pixel and text placement contol.

PIXEL SKETCH AND GRAPHICS EDITOR V2.0 only $19.95

KRUNCHER TS 2068 / TS 1000
From the Pacific Northwest comes one of the most exciting utilites ever written. KRUNCHER takes any BASIC program for the TS 2068 or TS 1000 and instantly reduces it to the tightest BASIC possible thereby conserving precious memory. Imagine all of those little memory saving tips developed over the years in one program which performs automatically and takes up less than 190 bytes! Simply load KRUNCHER, locate it where you want it, load or write your BASIC progrm, invoke KRUNCHER, blink your eye and it's done. Memory savings average 20-40%. Great learning aid for programmers of all levels.

KRUNCHER for TS1000 or TS2068 only $10.00
(Please specify TS1000 or TS2068)

TIME MACHINE TS 2068
The first SERIOUS COMPILER for the TS2068. Now you can convert BASIC programs to super fast MACHINE CODE without a lot of hassle. Converts both TS 2068 and SPECTRUM Programs. It functions as both an integer and floating point compiler simultaneously without the restrictions of either. Compiled code can be placed anywhere in RAM. Handles up to 27K of BASIC. Programs can be either written or loaded from tape. You've wainted a long time for this one and here it is!

TIME MACHINE for the TS2068 only $19.95

RIGTER JOYSTICK INTERFACE TS 1000
This is a software programmable Atari-type joystick interface. it can handle up to 16 different directions or commands easily. It has it's own self-contained memory so that it's software occupies no system ram. The software allows you to configure your joystick to ANY TS1000 game or graphic software (THRUST, for example) and it's ready to go. Rear expansion bus allows other peripherals and the interface is completely transparent.

RIGTER JOYSTICK INTERFACE for TS 1000
only $39.95

MINI XMOD TS 1000
Use your TS1000 and Westridge modem to up/download TS1000 software to any XMODEM BBS and see them run. Supports Memotech Centronics I/F and others for print-outs to full size printers. Standard RAM and Hunter Board versions included on same tape.

MINI XMOD for the TS 1000 only $20.00

LOADER V TS 2068
This program turns MTERM into a REAL communications program. Here's what you get. Autorepeat dialing, extra 20 number dialing directory, full TASWORD II and MSCRIPT text file handling capability, disk drive and Wafer drive compatible, multiple loading of Mterm's buffer while on line, and full XMODEM capability. This program is the COMPLETE LOADER SERIES.

LOADER V for TS 2068 only $10.00

CLONE TS 2068
A sophisticated program which allows the user to make back-up copies of ANY TS2068 or SPECTRUM software for their own use. Requires no fancy filters or extra tape recorders. Easy to follow instructions make it simple to protect your valuable originals.

CLONE for the TS2068 only $10.00

SPECIAL OFFER!!!
Save $5.00 when you order the combination of THRUST, RIGTER JOYSTICK INTERFACE, and KRUNCHER 1000 you pay only $64.95

SHIPPING INSTRUCTIONS Please enclose $2.00 shipping and handling with your order.

WEYMIL CORPORATION
BOX 5904
BELLINGHAM, WA 98227—5904